

**ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE LATACUNGA**

**FACULTAD DE INGENIERÍA EN SISTEMAS E
INFORMÁTICA**

**APLICABILIDAD DE LA TECNOLOGÍA WEB SERVICES BASADA
EN SOFTWARE OPEN SOURCE O DE CÓDIGO ABIERTO,
ORIENTADA A LA OFERTA ACADÉMICA DE LAS
UNIVERSIDADES DEL CENTRO DEL PAÍS**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

**HERRERA CHANCUSI JACKELINE SUSANA
TAPIA LEÓN MARIELA ALEXANDRA**

Latacunga, abril de 2005

CERTIFICACION

Se certifica que el presente trabajo fue desarrollado por las señoras: Jackeline Susana Herrera Chancusi y Mariela Alexandra Tapia León, bajo nuestra supervisión.

Ing. Raúl Rosero
DIRECTOR DE TESIS

Ing. José Luis Carrillo
CO DIRECTOR DE TESIS

“Tuve un sueño, en el que todas las máquinas se comunicaban entre sí...”. Imagínense un mundo que tuviera una red universal, inteligente, omnipresente y activa en todo momento, donde la posibilidad de un crash sea mínima porque no puede caerse toda simultáneamente, que cualquier aparato electrónico e informático se pudiera conectar y que cualquier usuario tuviera la posibilidad de hacer uso de sus recursos sin tener que configurar su equipo, porque éste es parte de dicha Red.

Sincera apología a Martin Luther King (1929 - 1968).

AGRADECIMIENTO:

Mi gratitud más sincera a Dios, el Supremo Hacedor, por su infinita bondad y misericordia para conmigo; a mis abnegados y amados Padres, que con amor, sacrificio y entrega total, siempre me apoyaron, en los buenos y difíciles momentos de mi vida y mi carrera, haciendo suyos mis sueños, ayudándome a cumplirlos de la mejor manera, y a forjar mi camino no solo como profesional, sino también como ser humano.

Mi reconocimiento a la ESPE sede Latacunga, noble Institución forjadora de grandes profesionales, que contribuyó en mi formación no solo Técnica y Científica, sino también humanista, a los distinguidos Docentes que en ella laboran, por compartir conmigo sus sabias y bastas enseñanzas; a la Asociación de Universidades y Escuelas Politécnicas Públicas del Ecuador (ASUEPPE), por haber auspiciado este proyecto de investigación; A mis queridos compañeros y amigos con quienes viví momentos inolvidables en esta etapa de mi vida.

Jackeline

Tengo que agradecer a Dios por permitirme compartir la alegría de un logro alcanzado con la gente que amo, estén aquí junto a mí o en tu infinita compañía. También debo agradecer a mi familia: a mi esposo y a mi hijo, por ser el motivo de mi vida y de lucha constante. A mis Padres y Hermanos, por ese apoyo y cariño que me brindan día a día. A mi Tío Gustavo, por su enorme amor y preocupación. A mis amigas y amigos de siempre, a los cuales estimo mucho. Y a la Escuela Politécnica del Ejército sede Latacunga, quien me forjó no solo como una profesional, si no como una persona que entiende el sentido de la palabra Humanidad.

Mariela.

DEDICATORIA:

Este logro lo dedico a los seres más importantes y sublimes de mi vida:

A Dios, a mis padres Susana y Víctor , a mi esposo Omar y a mi pequeño hijo Mateito, dignos símbolos de ternura, inspiración y amor, que iluminan mi existir y en quienes encuentro la motivación y aliciente para ser cada día mejor esposa, madre, hija y ser humano.

Jackeline

Dedico este proyecto a todas las personas que en los obstáculos ven oportunidades, y mientras van tras ellas, nunca dejan de lado lo más importante en la vida: el AMOR.

Mariela.

PRESENTACIÓN

El presente proyecto de investigación está enfocado a más de brindar un aporte científico a la comunidad informática, con lo que es la aplicabilidad de Web Services basada totalmente en XML y Software Libre, a orientar de mejor manera a los estudiantes que han concluido sus estudios secundarios y se encuentran en el gran dilema de: “Qué carrera quiero estudiar, a qué Universidad debo ir, en qué lugar”, cuestionamientos a los que nos hemos visto enfrentados, todos quienes hemos iniciado una carrera Universitaria y a los que se enfrentan y enfrentarán en algún momento los estudiantes que concluyen el bachillerato, por lo que ponemos en sus manos una herramienta en donde las Universidades y Escuelas Politécnicas conjuguen la información de su Oferta Académica y se sirvan de la tecnología para que los estudiantes tomen una mejor elección de acuerdo a la Universidad, especialidad, título, modalidad, ciudad, nivel, duración, escogiendo la que mejor se acople a sus necesidades.

CAPITULO I

APLICACIONES DISTRIBUIDAS

1.1. INTRODUCCIÓN

Sin duda alguna la evolución de la informática ha sucedido de una manera asombrosa. Resulta increíble imaginar que cincuenta años atrás las computadoras eran utilizadas únicamente por entidades o personas que podían costearlas y manejarlas, generalmente el Gobierno de los países desarrollados o gente muy acaudalada; hoy en día podemos encontrar una computadora a la vuelta de la esquina y a precios razonables.

Al principio los computadores por su tamaño y costo, funcionaban como una unidad completamente independiente; pero a partir de los años setenta con el surgimiento de los mini ordenadores se incrementa la necesidad de intercomunicación entre los mismos y aparecen las redes de área local LAN (Local Area Network, Red de Área Local) que permiten conectar decenas o hasta cientos de computadores dentro de un mismo lugar geográfico, un edificio por ejemplo, con el fin de transferir información

unas con otras, cuando este espacio resultó muy limitado surgen las redes de área extendida (Wide Area Network, Red de Área Extendida) que permiten conectar millones de máquinas alrededor de la Tierra dando paso lo que actualmente conocemos como la red de redes: Internet, que en un principio nos servía únicamente para comunicarnos vía mail y transferir archivos y ahora se ha convertido en un aliado poderoso en el desarrollo de empresas y negocios de toda índole.

1.2. APLICACIONES DISTRIBUIDAS

Las aplicaciones distribuidas son aquellas que están construidas sobre sistemas distribuidos por ello se realizará una introducción sobre este tema.

A partir de 1978 y hasta nuestros días muchos autores como Enslow, Coulouris, Bal, Schroeder, Tanenbaum han dado sus definiciones sobre Sistemas Distribuidos se toma para explicar el objetivo de los sistemas distribuidos la siguiente definición:

“Un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora”¹

Según Tanenbaum un sistema distribuido es aquél al que sus usuarios ven como un ordinario sistema centralizado, cuya diferencia radica en que la ejecución de los procesos se realiza en diferentes computadoras. La principal característica de un sistema distribuido es la transparencia; que significa que el uso de diversos procesadores deberá ser invisible o transparente al usuario. Otra forma de expresar esta misma idea es diciendo que el usuario verá al sistema distribuido como un uniprocador y no como una colección de computadoras diferentes

¹ Tomado del Libro “Sistemas Operativos Distribuidos” del autor Andrew S. Tanenbaum.

Todo sistema distribuido tiene tres características básicas:

Existencia de varios computadores. En general, cada uno con su propio procesador, memoria local, subsistema de entrada/salida y memoria.

Interconexión. Existen vías que permiten la comunicación entre los computadores, a través de las cuales pueden transmitir información ya sean estas redes alámbricas o inalámbricas.

Estado compartido. Los computadores cooperan para mantener algún tipo de estado compartido. Es decir, puede describirse el funcionamiento correcto del sistema como el mantenimiento de una serie de procesos globales que requiere la coordinación de varios computadores.

En cuanto a los campos de aplicación los sistemas distribuidos han sido desarrollados para realizar:

- Computación masivamente paralela, de propósito general y de alta velocidad.
- Ofrecer confianza y disponibilidad con Tolerancia a fallas.
- Respuesta a demandas con requisitos de tiempo real.

En síntesis, un sistema Distribuido no es más que un conjunto de computadores que se conectan entre sí mediante una red y se comunican mediante protocolos para coordinar sus actividades y compartir recursos, de manera totalmente transparente al usuario.

Los Sistemas Distribuidos surgen para brindar mayor eficiencia, haciendo posible que los procesos a ejecutarse se distribuyan a todos los procesadores que se encuentren libres (claro que la rapidez va a depender de la capacidad de los procesadores) realizándolo de forma transparente, así dan la idea al usuario de estar

trabajando en único computador aunque la información que procesan provenga o se almacene en otros puntos de red.

1.3. COMUNICACIÓN EN LOS SISTEMAS DISTRIBUIDOS

“La diferencia más importante entre un sistema distribuido y un sistema de un único procesador es la comunicación entre procesos”²

La comunicación entre procesos en sistemas con un único procesador se lleva a cabo mediante el uso de memoria compartida. En los sistemas distribuidos, al no haber conexión física entre las distintas memorias de los equipos, la comunicación se realiza mediante la transferencia de mensajes conocido como protocolos.

La ISO (International Standard Organization, Organización Internacional de Estándares) creó un modelo que ayuda a coordinar y entender los mensajes enviados por los computadores, este modelo se denomina MODELO DE REFERENCIA PARA INTERCONEXIÓN DE SISTEMAS ABIERTOS (**ISO OSI** o **modelo OSI**).

El modelo OSI establece los lineamientos para que el software y los dispositivos de diferentes fabricantes funcionen juntos. Posee 7 capas las mismas que están encargadas de una tarea en específico las capas del modelo OSI son:

² Tomado del Libro “Sistemas Operativos Distribuidos” del autor Andrew S. Tanenbaum.



Capas del Modelo OSI - Figura 1. 1

El modelo OSI comprende dos tipos de protocolos: los orientados hacia las conexiones y los protocolos sin conexión, los primeros precisan establecer de antemano una conexión y al finalizar el intercambio de datos cerrarla como el caso del teléfono y los segundo no requieren de este paso y los mensajes se transmiten en forma de datagramas que consiste en “un único mensaje, independiente y en un solo sentido, enviado desde un programa de aplicación a otro”³

1.4. CARACTERÍSTICAS PRINCIPALES DE UN SISTEMA DISTRIBUIDO.

Los sistemas distribuidos surgieron para satisfacer una esencial condición humana: EL DERRIBAR LAS FRONTERAS PARA ALCANZAR SUS METAS. Hoy en día

³ Tomado del Libro “Comunicaciones y Redes de Procesamientos de Datos” del autor Nestor González Sainz.

existen negocios que gracias a empresarios decididos a derribar sus fronteras han extendido sus dominios en varios países; pero no han ido solos, la tecnología de las comunicaciones han sido su principal aliado, ayudándolos a alcanzar más eficientemente sus objetivos. Estos negocios que se hallan dispersos dentro de un país a alrededor del mundo requieren de sistemas distribuidos que les permitan manejar y controlar sus empresas desde cualquier lugar del globo con total confianza y seguridad, por esto un sistema distribuido debe poseer:

- Seguridad.
- Confiabilidad
- Escalabilidad.
- Tolerancia a Fallos.
- Compartimiento de recursos.
- Y ser un sistema abierto.

Al reunir estas características podemos hablar de TRANSPARENCIA esencia de las aplicaciones distribuidas. Mediante la transparencia se consigue que un usuario no perciba los fallos que pueden ocurrir en la red, o no note que la red ha crecido mientras él trabajaba o incluso, permite que el usuario comparta recursos sin necesariamente saber con quién y en dónde. Cuando un sistema distribuido cumple con estos aspectos, los programadores de APLICACIONES DISTRIBUIDAS deben despreocuparse de esos detalles y tan solo hacer uso de los recursos que los sistemas distribuidos les ofrecen.

Se aclara que las características anteriormente descritas son deseables en un sistema distribuido, esto quiere decir que no necesariamente debe reunir todas estas características para ser un sistema distribuido, dependiendo de cuantas características reúna podemos hablar de sistemas más o menos distribuidos.

1.4.1. SEGURIDAD

Todo sistema tanto centralizado como distribuido es vulnerable en lo que ha seguridad se refiere, aunque en un sistema centralizado parezca más sencillo controlar la seguridad porque se trata de un solo computador, puede ser que para un intruso le resulte más fácil violar la seguridad en un sistema centralizado que en uno distribuido, ya que en éste la información se halla bajo la administración y control de distintos dominios y administradores complicando el trabajo de acceso a datos a personas no autorizadas.

En un sistema con un procesador, el usuario realiza un proceso de autenticación, a partir de ese momento el sistema sabe quién es el usuario y puede verificar la validez de cada intento de acceso.

En un sistema distribuido cuando llega un mensaje a un servidor con cierta solicitud, le resulta difícil determinar de que cliente proviene. El sistema no puede confiar en un nombre o campo de identificación en el mensaje, puesto que el emisor puede mentir.

1.4.2. CONFIABILIDAD

Uno de los objetivos principales de la construcción de sistemas distribuidos fue el hacerlos más confiables que los sistemas con un único procesador, cuya labor es encargarse del trabajo de algún computador que por cualquier circunstancia no funciona como es debido. Por esto los sistemas distribuidos cuentan con cierto número de servidores para que todo funcione.

Cuando se habla de confiabilidad también se habla de disponibilidad que se refiere “a la fracción de tiempo en que se puede utilizar el sistema, una herramienta para mejorar la disponibilidad es la redundancia: se puede duplicar piezas clave del

hardware y del software, de modo que si una de ellas falla, las otras puedan llenar su hueco”.⁴

1.4.3. ESCALABILIDAD

Cuando surge una empresa nueva por lo general ésta es pequeña de tamaño pero grande en ambición por lo que se plantea como objetivo ir escalando peldaño a peldaño en el mundo de los negocios hasta alcanzar un importante sitio. Cuando una empresa crece, también crecen sus necesidades; puede ser que al principio hayan empezado con un solo computador y una impresora, pero poco a poco y ha medida del crecimiento de la empresa formaron una pequeña red LAN dentro de su negocio. Pero nuestra empresa no termina aquí, el buen manejo del negocio les permitió ampliarse y convertirse en una multinacional con lo que su red de área local se conectó con otras redes formando una red WAN. Desde luego que dicha empresa no hubiera alcanzado tal dimensión si hubiese tenido que pasar por miles de problemas cada vez que se aumentaba un computador a la red.

La escalabilidad en los sistemas distribuidos evita este tipo de problemas, podemos aumentar tanto hardware o software a un sistema conservando su característica principal: la transparencia, así el usuario podrá seguir trabajando normalmente mientras su red y su negocio se expande.

1.4.4. TOLERANCIA A FALLAS

Ocasionalmente los computadores están expuestos a fallas tanto en hardware como en software, por lo tanto se debe construir sistemas con tolerancia a fallas mediante procesos que puedan recuperarse de las mismas, afortunadamente los sistemas distribuidos intrínsecamente son tolerantes a fallas ya que en el caso de existir un

⁴ Tomado del Libro “Sistemas Operativos Distribuidos” del autor Andrew S. Tanenbaum.

indebido comportamiento del sistema sólo se verán afectados los usuarios de ese computador los mismos que incluso pueden cambiarse (si es posible) a otro computador y seguir trabajando con total normalidad mientras se busca la solución a la falla ocasionada.

Debemos tener en cuenta que los computadores con información relevante para los usuarios de toda la red en su conjunto deben ser replicados para que el sistema sea tolerante a fallas.

1.4.5. COMPARTIMIENTO DE RECURSOS (MODELO CLIENTE-SERVIDOR)

Una de las principales características de los sistemas distribuidos es la capacidad de ofrecer y utilizar recursos tanto físicos como lógicos en una red es decir el compartimiento de recursos. El esquema más utilizado para realizar estas transacciones es el modelo CLIENTE – SERVIDOR, Donde el **Servidor** es aquel que gestiona las peticiones a procesar, dicho de otra forma es el que brinda el servicio y un **Cliente** son los procesos o datos enviados por la misma máquina o por máquinas remotas al servidor para su debido procesamiento o almacenaje, es decir es el que solicita el servicio.

El servicio puede ser la ejecución de un determinado algoritmo, el acceso a determinado banco de información o el acceso a un dispositivo hardware.

El protocolo de comunicación entre un modelo cliente – servidor es como sigue: En primer lugar el cliente realiza una petición de servicio al servidor el cual procesa la petición y finalmente devuelve una respuesta a la petición del cliente, o un mensaje de error en caso de no haber podido ejecutarlo correctamente.

Cabe indicar que al mencionar a un cliente o a un servidor no se está refiriendo a las máquinas o estaciones de trabajo, sino a los procesos que son los que se

comportarán como clientes o servidores, esto dependiendo si está utilizando u ofreciendo, es por esta razón que un mismo computador en un momento dado se convierte en cliente y servidor a la vez.

“Por extensión, se puede aplicar el esquema cliente-servidor dentro de una misma máquina, donde el proceso servidor y el proceso cliente son dos procesos independientes que corren dentro de la misma instancia de sistema operativo.”⁵

Los sistemas Cliente - Servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece:

1.4.5.1 Representación distribuida

La interacción con el usuario se realiza básicamente en el servidor. El cliente hace de pasarela, de sistema de acceso a los elementos hardware pantalla y teclado.

1.4.5.2 Representación remota

Los datos se envían sin formatear, y es el cliente el responsable de formatear los datos y realizar las acciones de interacción con el usuario. En este caso, la aplicación y la base de datos se encuentran en el servidor.

1.4.5.3 Lógica distribuida

En el cliente se llevan a cabo la interacción con el usuario y la parte más trivial de la lógica de la aplicación. En este caso, se llevan a cabo controles básicos de rango de campos, campos obligatorios, etc, mientras que el bloque de la lógica permanece en el servidor.

⁵ Fuente de consulta: <http://dmi.uib.es/~bbuades/sistdistr/sistdistr.ppt>.

1.4.5.4 Gestión Remota de Datos

Tanto la interacción con el usuario como la aplicación residen en el cliente, siendo el servidor únicamente el depositario de los datos.

1.4.5.5 Base de Datos Distribuidas

El cliente debe conocer la topología de la red, así como la disposición y ubicación de los datos. En este caso, se delega parte de la gestión de base de datos a los clientes.

1.4.5.6 Cliente servidor a tres niveles (three tier)

La aplicación se distribuye en los tres niveles: aplicación, datos e interface de usuario.

1.4.6. SISTEMA ABIERTO

La idea básica de un sistema abierto es que pueda ser fácilmente ampliado o reducido de acuerdo a las necesidades sin dependencia alguna de su fabricante o programador.

Existen sistemas abiertos tanto en Hardware como en Software. En el primer caso se trata de un sistema abierto cuando nos permite añadir nuevos dispositivos electrónicos sin que estos sean necesariamente originarios del constructor o fabricante y de igual forma para el caso de los sistemas abiertos respecto al software, la adición de nuevos programas debe ser realizada de forma sencilla sin entrar en conflictos entre las diferentes empresas de software.

Los sistemas distribuidos y aplicaciones distribuidas deben ser concebidos como un sistema abierto que se adapte a cualquier tipo de arquitectura y sistema operativo de tal forma que pueda ser ampliado a distintos computadores y ofrecer nuevos servicios que beneficien a la sociedad.

1.4.7. TRANSPARENCIA

Como se ha expuesto la principal característica de un sistema distribuido es la transparencia, con la cual se pretende ocultar toda la compleja red y conceptos que se esconde detrás de los sistemas distribuidos y mostrar al usuario un sistema como uno más de tiempo compartido de un solo procesador con los que ya ha venido trabajando.

Tipos de transparencia respecto a:

1.4.7.1 Transparencia de Acceso

Permite que el acceso a los objetos locales o remotos se realicen de idéntica forma.

1.4.7.2 Transparencia de Localización

El usuario puede acceder a los recursos de cualquier computador sin conocer previamente su ubicación, es decir no necesitan conocer su dirección.

1.4.7.3 Transparencia de Fallos

Los fallos no son percibidos por los usuarios por lo que pueden seguir trabajando con total normalidad.

1.4.7.4 Transparencia de Réplica

Se pueden utilizar varias instancias de objetos para mejorar la fiabilidad y rendimiento del sistema.

1.4.7.5 Transparencia de Migración

Significa que los recursos puedan moverse de una posición a otra sin tener que cambiar sus nombres

1.4.7.6 Transparencia de Paralelismo

Permite que varios usuarios realicen actividades en paralelo.

1.4.7.7 Transparencia de Escala

Ayuda a que las aplicaciones o componentes del sistema aumenten sin cambiar la estructura del sistema ni afectar las actividades de los usuarios.

1.5. MODELOS DE LAS APLICACIONES DISTRIBUIDAS

1.5.1. PROCESAMIENTO DISTRIBUIDO BASADO EN ENTRADA Y SALIDA.

El procesamiento distribuidos basado en entrada y salida, permite comunicarse con un proceso remoto de manera similar a leer o escribir en un archivo. Este procesamiento distribuido hace uso de un mecanismo denominado SOCKETS que permite emitir y recibir información entre distintos computadores.

La comunicación entre procesos a través de sockets se basa en la filosofía CLIENTE-SERVIDOR, los mismos que se hallan en la capa de transporte del Modelo OSI.

Para establecer una conexión mediante sockets es necesario tener en cuenta dos características: la familia de sockets que se está utilizando, y el tipo de conexión.

La **familia** de sockets indica el tipo de direcciones que se utilizarán para especificar las diferentes máquinas. Las más comunes son la familia de protocolos internos de Unix, que se utiliza para comunicar procesos que se ejecutan en la mismo computador y la familia de protocolos Internet, que se utiliza para la transmisión de protocolos como el TCP y el UDP.

El **tipo de conexión** puede ser orientado a conexión o no orientado a conexión (Tipo datagrama). En el primer caso debe existir una conexión permanente entre los dos computadores, de esta manera una vez que se haya realizado la conexión todos los mensajes recorrerán el circuito establecido. En el segundo caso no se crea una conexión permanente, esto implica que los mensajes pueden seguir rutas distintas y no se garantiza que lleguen en el mismo orden en que fueron enviados.

Este tipo de procesamiento es considerado de bajo nivel ya que el programador debe preocuparse de establecer los protocolos de comunicaciones, de la representación de datos, entre otros.

1.5.2. PROCESAMIENTO DISTRIBUIDO BASADO EN LLAMADAS A PROCEDIMIENTOS REMOTOS.

En el procesamiento distribuido basado en llamadas a procedimientos remotos conocido por sus siglas en Inglés como RPC (Remote Call Procedures, Llamada a Procedimientos Remotos) permite comunicarse con un proceso remoto de forma similar a invocar un procedimiento dentro de un programa, con la diferencia que el

procedimiento puede estar ubicado en otro computador. Normalmente, el computador que hace la petición es referenciado como cliente y el que responde a una petición será el servidor.

Posee un enfoque de nivel más alto que el orientado a entrada/salida y los detalles de enviar y recibir mensajes quedan ocultos al programador.

El paquete RPC fue desarrollado por Sun Microsystems, y es una colección de herramientas y funciones de librería que se las ha declarado como estándares.

RPC es una API (Application Program Interface, Interfaz de Programación de Aplicación) disponible para el desarrollo de aplicaciones distribuidas el mismo que permite que los programas llamen a procedimientos o subrutinas que se ejecutan en un computador remoto. El Cliente envía un mensaje de petición al Servidor y espera por un mensaje de respuesta. La llamada incluye los parámetros del procedimiento y la respuesta los resultados. Para que la transferencia sea independiente de la plataforma, todos los datos intercambiados entre el cliente y el servidor son convertidos al formato denominado XDR (eXternal Data Representation, Representación de Datos Externos) por el servidor, y convertidos otra vez a la representación del computador local por el receptor. Como las aplicaciones RPC no usan puertos reservados, no existe la seguridad de que el puerto originalmente pensado para ser usado en una determinada aplicación no haya sido utilizado por algún otro proceso. Por lo tanto, las aplicaciones RPC escogen cualquier puerto que puedan utilizar, y lo registran con el denominado demonio mapeador de puertos que actúa como un distribuidor de servicios, pero se corre el riesgo de que cuando el mapeador de puertos falla, toda la información de puertos RPC se pierde; lo que implica que hay que reiniciar todos los servidores RPC manualmente, o reiniciar todo el computador.

Para implementar RPC, existen algunas dificultades que deberían ser tratadas:

- El cliente deberá localizar el servidor y verificar que esté disponible.
- El cliente y el servidor deberán convenir que el procedimiento que se invocará está disponible en el servidor, y que la especificación de ambos computadores es compatible.
- Los argumentos para el procedimiento remoto debe ser convertido en un formato de arquitectura neutral y transferidos en la red al servidor.
- Una vez que el procedimiento remoto se completó, cada resultado debe ser transferido de nuevo al cliente. Necesitan también ser convertidos de y desde un formato de arquitectura neutral.
- El sistema deberá asegurarse de la conexión de red.
- Debe verificar cualquier problema de RPC del cliente y el servidor, aunque sea muy pequeño.
- Por último, pero no menos importante, el servidor necesita un mecanismo de seguridad para confirmar que el cliente ejecute el procedimiento en el servidor.

Muchos protocolos pueden ser usados por RPC, incluyendo NetBEUI, IPX/SPX y otros, pero ambos, cliente y servidor deben elegir el mismo protocolo. Un cliente usando RPC sobre NetBEUI no puede comunicarse a un servidor con RPC sobre TCP/IP, aunque un servidor puede soportar múltiples protocolos simultáneamente. El protocolo más comúnmente usado para soportar RPC's es el TCP/IP.

La ventaja de este tipo de procesamiento respecto al anterior es que el programador no debe preocuparse de establecer los protocolos de comunicaciones, esto es transparente para el programador.

1.5.3. PROCESAMIENTO DISTRIBUIDO BASADO EN OBJETOS DISTRIBUIDOS.

“Los sistemas distribuidos requieren que las partes que los componen y que se ejecutan en diferentes espacios de direcciones (posiblemente en diferentes máquinas), tengan la capacidad de comunicarse entre sí. “⁶

Una de las primeras soluciones para la comunicación entre dos o más computadores fueron los sockets, que tienen la capacidad de comunicar dos procesos, ya sea mediante datagramas o flujos de datos (streams). Pero los sockets requieren que las aplicaciones declaren sus propios protocolos para codificar y decodificar los mensajes que intercambian, lo que vuelve más complejo la labor del programador para intercambiar datos y aumenta la posibilidad de errores durante la ejecución.

Posterior al empleo de los sockets y basadas en ellos aparecen las Llamadas a Procedimientos Remotos (RPC) donde la comunicación entre los elementos que componen el sistema distribuido, se realiza mediante la invocación de funciones que se encuentran en espacios de direcciones diferentes. En este caso, el programador tiene la impresión de trabajar con procedimientos locales, mientras que en realidad el sistema RPC se encarga de empaquetar los argumentos y enviarlos al proceso que contiene el código que implementa a la rutina remota. Los sistemas codifican los parámetros de la invocación, así como los valores de vuelta en una representación externa de los datos. Un ejemplo de este tipo de representaciones externas es XDR (eXternal Data Representation).

Debido a que las tecnologías orientadas a objetos se han consolidado como una de las herramientas más eficaces en el desarrollo de software es de esperar que los

⁶ Fuente de consulta: <http://www.revista.unam.mx/vol.2/num1/art3/>

sistemas distribuidos hagan uso de esta tecnología ya que en el paradigma basado en objetos el estado de un programa ya se encuentra distribuido de manera lógica en diferentes objetos, lo que hace a la distribución física de estos objetos en diferentes procesos o computadoras una extensión natural.

En los lenguajes cuyo paradigma de programación no es el procedimental, sino el orientado a objetos, se requiere ya no invocar procedimientos remotos, sino a métodos de objetos remotos. El empleo de objetos distribuidos, en lugar de procedimientos remotos, implica varias ventajas como la orientación a objetos, movilidad de las aplicaciones, los patrones de diseño, la seguridad, etcétera.

La principal desventaja de los objetos distribuidos, con respecto a las llamadas a procedimientos remotos y Sockets, es el rendimiento.

1.6. VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES DISTRIBUIDAS.

1.6.1. VENTAJAS DE LAS APLICACIONES DISTRIBUIDAS.

Para explicar las ventajas que brindan las aplicaciones distribuidas, se incluye a continuación un ejemplo para entender sus beneficios y comprender porque este tipo de aplicaciones forman parte del actual ritmo de vida.

Imagínese que posee un supermercado con varias tiendas dispersas a nivel nacional del cual usted es el Gerente, cada tienda realiza sus ventas y mantiene su inventario de manera local en una computadora local, pero qué tal si usted en un momento determinado necesita conocer ¿Cuántas cajas de manzanas posee en todas sus tiendas?.

Para realizar esta consulta usted debe solicitar a cada uno de los administradores de las tiendas del supermercado un informe con el número de cajas de manzanas que

posean en su tienda. Esto tomará de horas a días dependiendo de la manera de enviar los informes, vía fax o personalmente. Lo cierto es que al realizar este proceso se ha incurrido en una pérdida innecesaria de tiempo, lo cual una aplicación distribuida permite mitigar, haciendo que usted obtenga respuestas a sus consultas en tiempo real mejorando el proceso de toma de decisiones.

Las aplicaciones distribuidas también permiten ahorrar dinero. Si retoma el ejemplo del supermercado, suponga que ahora requiere un reporte con distinta información acerca de la administración de cada una de las tiendas, para al final obtener un reporte global del estado financiero en el que se encuentra el supermercado. Para este trabajo los administradores debían reunirse para estructurar dicho reporte, lo cual significaba buscar un día para realizar la reunión, trasladarse a un lugar acordado, gastar dinero en movilización, estadía y alimentación para finalmente aunar la información de cada tienda y obtener el reporte. Este es otra de las ventajas que le ofrece las aplicaciones distribuidas, ya que permiten no sólo ahorrar tiempo en la toma de decisiones si no también dinero, en vista de que toda la información que requiere se puede hallar desde su propio computador.

Si por cualquier razón el computador de alguna tienda se daña, o se infecta de un virus, seguramente que la información que poseía se perderá sin remedio, y estancará el trabajo conjunto del resto de tiendas causándole grandes inconvenientes, los sistemas distribuidos le permiten poseer réplicas de la información en otro computador brindándole mayor confianza, de esta manera el resto de usuarios podrán seguir trabajando con total normalidad mientras se arregla el computador descompuesto.

El crecimiento al que por lo general están expuestas las empresas obligan también a crecer respecto a los computadores. Resultaría problemático el detener las operaciones de la empresa sólo por que se está agregando un nuevo computador a

la red, con un sistema distribuido se puede añadir tanto hardware como software sin obstaculizar en forma alguna en las operaciones de la empresa.

Es decir que los sistemas distribuidos y las aplicaciones distribuidas permiten compartir información sin importar las distancias y ahorrando tiempo y dinero.

1.6.2. DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS

A pesar de las grandes ventajas que nos ofrecen los sistemas distribuidos existen también algunas desventajas, entre las principales se encuentra el software del cual se conoce poco sobre su diseño, implantación y uso, principalmente se desconoce el sistema operativo sobre el cual deben funcionar y el lenguaje sobre el cual va a ser desarrollado.

Otro de los problemas que debe enfrentar un sistema distribuido es la red, porque le puede restar seguridad y confianza al usuario debido a la pérdida de mensajes o también puede ocurrir que se sature la red disminuyendo la velocidad de tránsito.

Pero a pesar de los inconvenientes que puede involucrar el trabajar con una aplicación distribuida, es preferible arriesgarse un poco antes que seguir incomunicado con el resto de personas: socios o clientes y gastar dinero en reuniones innecesarias.

1.7. ASPECTOS A TENER EN CUENTA EN EL PROCESO DE INGENIERÍA

Algunas de las primeras y más importantes cosas a tomar en cuenta en el proceso de ingeniería del sistema distribuido, son:

1.7.1. PROTOCOLOS DE COMUNICACIONES

Los protocolos de comunicaciones son de cierta manera más importantes que la propia arquitectura distribuida o centralizada. “Un buen protocolo permite que se pueda pasar, sin un coste adicional de rediseño o codificación, de una arquitectura centralizada a una distribuida, y viceversa.”⁷

1.7.2. MIDDLEWARE

Es la herramienta o conjunto de herramientas que nos permitirán gestionar y coordinar los mecanismos de comunicación entre dos computadores.

Los middleware permiten independizar el servicio y su implementación, del sistema operativo y de los protocolos de comunicaciones, además que permite la convivencia de distintos servicios en un mismo computador.

Un middleware “Es un sistema abierto independiente del fabricante que no depende del hardware y del sistema operativo subyacente.”⁸

Cada casa desarrolladora de software pugna por sacar al mercado una herramienta middleware que satisfaga a sus clientes, entre estas tenemos: DCE de Open Group, CORBA de OMG, RMI de Java, DCOM de Microsoft, entre otros.

A continuación se da a conocer de una manera general los aspectos más importantes de algunas de estas herramientas.

1.7.2.1 DCE (Distributed Computing Environment, Entorno de Informática Distribuida)

⁷ Fuente de consulta: <http://dmi.uib.es/~bbuades/sistdistr/sistdistr.ppt>.

⁸ Fuente de consulta: <http://www.geocities.com/hsagastegui2002/TrabajoCorba1.html>

Fue creado por Open Systems Foundation integrada por X/Open, y Open Group. DCE tiene sus inicios desde 1992.

“Se trata de un conjunto integrado de servicios de sistema que proporcionan un entorno distribuido flexible cuyo primer objetivo es resolver los problemas de interoperabilidad de los entornos heterogéneos interconectados.

DCE soporta la construcción e integración de aplicaciones cliente/servidor, intentando ocultar al usuario la complejidad inherente al proceso distribuido”⁹

DCE soporta los estándares OSI como los protocolos de transporte y de red TCP/IP, DNS (Domain Name Service, Servicio de Nombre de Dominio) y NTP (Network Time Protocol, Protocolo de Tiempo de Red).

1.7.2.2 CORBA (Common Object Request Broker Architecture)

Es una tecnología creada y mantenida por OMG (Object Management Group), un consorcio formado por un grupo de empresas implicadas en el uso de programación distribuida, desde 1989.

La principal característica de CORBA es su independencia del lenguaje de programación mediante el uso de una interfaz de programación común denominada IDL (CORBA Interface Definition Language, Interfase de Definición de Lenguaje). Los objetos CORBA difieren de los objetos de programación típica en tres aspectos:

- Los objetos CORBA pueden correr en cualquier plataforma.
- Los objetos CORBA pueden estar localizados en cualquier lugar de la red.

⁹ Fuente de consulta: <http://www.inst-informatica.pt/v20/ibst/estandar/8/ibdce.htm>

- Los objetos CORBA pueden ser escritos en cualquier lenguaje que tenga un mapeado IDL.

1.7.2.3 RMI (Remote Method Invocation, Invocación de Métodos Remotos)

Es el mecanismo ofrecido por Java que permite que un procedimiento o método pueda ser invocado remotamente.

RMI, forma parte de todo JDK(Java Development Kit, Paquete Desarrollador en Java) , por ende, cualquier plataforma que tenga acceso a un JDK también tendrá acceso a estos procedimientos, esto puede ser visto como una ventaja o desventaja ya que RMI permite un solo lenguaje para la Interoperabilidad, JAVA.

La conectividad entre las funciones remotas, al igual que cualquier otra comunicación de Red se lleva acabo mediante un puerto TCP/IP. Esta comunicación requiere definir lo que es denominado RMI Registry (RMI Registro), el RMI Registry se establece en un puerto TCP/IP que está designado por defecto el 1099 y mantiene un mapa de las funciones remotas que serán utilizadas.

RMI proporciona comunicación remota entre programas escritos en Java. Si uno de los programas está escrito en otro lenguaje, se debe considerar la utilización de IDL al igual que CORBA, para trabajar.

1.7.2.4 DCOM (Distributed COM)

Fue desarrollado por Microsoft y es una extensión de COM (Component Object Model, Modelo de Componentes de Objetos) cuyo principal objetivo como el de los anteriores es la comunicación entre objetos entre computadores distintos, en una LAN, WAN, o incluso en Internet.

“DCOM es una extensión de COM, y éste define como los componentes y sus clientes interactúan entre sí. Esta interacción es definida de tal manera que el cliente y el componente puede conectar sin la necesidad de un sistema intermedio. El cliente llama a los métodos del componente sin tener que preocuparse de niveles más complejos.”¹⁰

DCOM es completamente independiente del lenguaje. Java, Microsoft Visual C++, Microsoft Visual Basic, Delphi, PowerBuilder, y Micro Focus COBOL interactúan perfectamente con DCOM.

Con la independencia de lenguaje de DCOM, los desarrolladores de aplicaciones pueden elegir las herramientas y lenguajes con los que estén más familiarizados. La independencia del lenguaje permite crear componentes en lenguajes de nivel superior como Microsoft Visual Basic, y después reimplementarlos en distintos lenguajes como C++ o Java.

DCOM proporciona transparencia o es independiente a la infraestructura de la red: DCOM puede utilizar cualquier protocolo de transporte, como TCP/IP, UDP, IPX/SPX y NetBIOS. proporcionando un marco de seguridad a todos estos protocolos.

1.7.3. FASE DE ANÁLISIS.

Prácticamente no hay diferencias respecto a un sistema informático tradicional. Se debe definir el costo en comunicaciones que puede asumir la organización.

1.7.4. FASE DE DISEÑO.

¹⁰ <http://www.elai.upm.es/spain/Investiga/GCII/areas/administracion/DCOM.htm#Introducción>

El diseño de entidades, en raras ocasiones se verán afectadas ya que aparecerán nuevos conjuntos de datos que no deberán ser vistas como nuevas entidades, sino como información que debe viajar entre nodos.

Respecto al diseño de tablas, se debe especificar su implementación:

- Desde qué nodos debe ser accesible
- Qué nivel de acceso se precisa desde cada uno de ellos, cómo implementarlo.

1.7.5. IMPLEMENTACIÓN DE BASE DE DATOS DISTRIBUIDAS.

No hay entornos puramente distribuidos. Debe analizarse, tabla a tabla, qué distribuir, qué centralizar y cómo hacerlo.

1.7.6. FASE DE PRUEBAS.

Debido a la complejidad del sistema, serán necesarias varias fases:

- Pruebas de funcionalidad de la aplicación. Se puede llevar a cabo sobre computadores de desarrollo y estaciones de trabajo de forma paralela.
- Pruebas de carga del servidor.
- Pruebas de integridad de datos. Son especialmente importantes en el caso de bases de datos distribuidas.
- Pruebas transaccionales.
- Pruebas de red.

1.8. APLICACIONES DE LOS SISTEMAS DISTRIBUIDOS

Las aplicaciones más comunes de los sistemas distribuidos son aquellas con las que ya hemos trabajado habitualmente el Correo Electrónico y la Transferencia de Archivos.

Comercialmente los Sistemas Distribuidos han sido desarrollados en su mayor parte para las Agencias de Viajes y las Líneas Aéreas. De igual manera la Banca se ha beneficiado mediante el uso de los sistemas distribuidos ampliando su cobertura a través de Cajeros Automáticos dispersos a escala nacional.

El mundo de la Multimedia también ha alcanzado gran apogeo gracias a los sistemas distribuidos este es el caso de las videoconferencias y de los juegos multiusuarios.

La educación ha distancia ahora se ha vuelto imperceptible gracias a que las aplicaciones distribuidas permiten que tanto estudiantes como profesores se comuniquen mediante clases virtuales, video conferencias, chat room, foros, entre otros.

CAPÍTULO II

SOFTWARE LIBRE U OPEN SOURCE

2.1. INTRODUCCIÓN

En la actualidad Internet es una de las más poderosas e importantes herramientas de comunicación e intercambio de información, la mayoría de los sitios web que aquí existen, así como los correos electrónicos que a diario son intercambiados en la Red, son soportados por programas Open Source (de código abierto), incluso, el importante sistema de dominios que permite referirse a un servidor de Internet con palabras y no con números, utiliza Open Source.

2.2. ORÍGENES

El Open Source es el resultado del movimiento "Open Source Initiative" (OSI), pero en la actualidad se prefiere el término "Software Libre" en lugar de Open Source, por su definición más amplia y clara , utilizando además su traducción en Inglés "free software", teniendo en cuenta que free significa libre y no gratis, es decir software libre se refiere a "libertad de expresión" y no a precio.

El software libre apareció en la década de los 70's encabezando su propulsión Richard Stallman.

"El modelo de Software Libre radica en el deseo de ayudar y ser ayudado. Por ejemplo, al escribir una novela, un escritor puede sentirse confundido y necesita ayuda, entonces le deja leer a sus personas cercanas un capítulo para que le

ayuden, en el proceso sus amigos aportan ideas e inclusive le ayudan a corregir su trabajo y es más, si su temática es atractiva le ayudan a escribir un nuevo segmento. “¹¹

En el mundo del desarrollo informático sucede algo similar, una persona escribe un programa para resolver una necesidad específica, la comenta con otros y estos al ver que dicho programa también los beneficia, le ayudan a realizarlo. Mientras esto ocurre, la propiedad intelectual no se vuelve una lucha de egos, se transforma en un trabajo de todos y para todos; se generan líderes naturales con autoridad moral sobre el proyecto, pero jamás dueños.

2.3. DEFINICIÓN DE SOFTWARE LIBRE

El Software libre es un software gratuito o comercial, que permite que los programadores puedan leer, modificar y redistribuir el código fuente de un programa de una forma libre, es decir sin estar sujetos a las restricciones que implica el uso de software propietario.

2.3.1. LIBERTADES DEL SOFTWARE LIBRE

2.3.1.1 Libertad 0

Libertad de usar el programa con cualquier propósito, es decir emplearlo o utilizarlo en algún sistema informático o tipo de trabajo que se requiera sin comunicarle a nadie.

2.3.1.2 Libertad 1

¹¹ Fuente de consulta: <http://manifiesto.cofradia.org/introduccion.html>

Libertad de investigar cómo funciona el programa para poder adaptarlo a las necesidades de quien lo está estudiando sin ninguna restricción.

2.3.1.3 Libertad 2

Libertad de distribuir copias a cualquier persona y en el lugar que se desee, ya sea gratis o cobrando por él, esto quiere decir que no se debe pedir o pagar permisos a nadie para distribuirlo, debiendo incluir el código fuente y el ejecutable o programa binario, aunque algunos lenguajes no permiten la creación de archivos ejecutables, pero si se encuentra la manera de crearlos se los debe incluir.

2.3.1.4 Libertad 3

Libertad de modificar el programa para mejorarlo y utilizarlo de manera privada si así se lo desea, sin tener que avisar a nadie y si se lo publica para beneficio de otros usuarios, no se le debe pedir permiso a nadie.

2.3.2. EL PROYECTO GNU

El proyecto GNU consiste en el desarrollo de un sistema operativo y juego de aplicaciones totalmente libre y compatible con UNIX. El proyecto incluye desarrollar una versión libre de cualquier aplicación que no se disponga libre.

Este proyecto no está limitado a sistemas operativos existiendo también aplicaciones como juegos, procesadores de texto, hojas de cálculo, etc.

2.3.2.1 Sistema GNU

El sistema GNU es un sistema operativo libre completo estilo Unix desarrollado por el Proyecto GNU.

Se puede o no pagar un precio por obtener software GNU. Una vez que se obtiene éste, se goza de tres libertades específicas para usarlo:

- La libertad de copiar el programa y redistribuirlo.
- La libertad de cambiar el programa como se desee, por tener acceso completo al código fuente.
- La libertad de distribuir una versión mejorada ayudando así a construir la comunidad pudiendo o no cobrar una cuota por el acto físico de efectuar la copia.
- Debido a que la razón de GNU es ser libre, cada componente individual en el sistema GNU tiene que ser software libre. No todos tienen que estar protegidos con copyleft (regla de distribución de software libre), sin embargo; cualquier tipo de software libre es legalmente apto de incluirse si ayuda a alcanzar metas técnicas.

2.3.2.2 Software GNU

El software GNU es un software bajo el auspicio del Proyecto GNU.

La mayoría del software GNU está protegido con copyleft; sin embargo, todo el software GNU debe ser software libre.

2.3.3. FUNDACIÓN PARA EL SOFTWARE LIBRE (FSF)

La Fundación para el Software Libre está dedicada a eliminar las restricciones sobre el copiado, redistribución, entendimiento, y modificación de programas de computadoras, promocionando el desarrollo y uso del software libre en todas las

áreas de la computación, pero muy particularmente ayudando a desarrollar y distribuir software GNU. La FSF sigue la regla de no instalar ningún programa propietario, excepto temporalmente para el propósito específico de escribir un reemplazo libre para dicho programa.

2.3.4. DISTRIBUCIÓN DE SOFTWARE LIBRE

Independientemente de cómo se haya conseguido el software libre, ya sea por un importe económico o de forma gratuita, siempre está vigente la libertad de copiarlo y modificarlo, e incluso vender copias de él.

Existen ciertos tipos de reglas sobre la manera de distribuir software libre como por ejemplo:

COPYLEFT.- (izquierdo de copia) Esta regla significa que cuando se redistribuya un programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

2.3.5. CATEGORÍAS DEL SOFTWARE LIBRE

2.3.5.1 Software de Fuente Abierta

“El término software de fuente abierta es usado por algunas personas para dar a entender más o menos lo mismo que software libre”¹²

¹² Fuente de consulta: www.gnu.org/philosophy/categories.es.html. Free Software Foundation.1999

2.3.5.2 Software de Dominio Público

El software de Dominio Público no está protegido con copyright, pero es un caso especial de software libre no protegido con copyleft, que quiere decir que algunas copias o versiones modificadas no pueden ser libres completamente.

2.3.5.3 Software protegido con copyleft

El software protegido con copyleft es software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional al momento de redistribuir o modificar el software. Esto quiere decir que cada copia del software, aun si ha sido modificado, debe ser software libre.

2.3.5.4 Software abarcado por GPL (**General Public License/Licencia Pública General**).

La GPL de GNU es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El Proyecto GNU la utiliza como los términos de distribución para la mayoría del software GNU.

2.3.6. VENTAJAS Y DESVENTAJAS DE UTILIZAR SOFTWARE LIBRE

2.3.6.1 Ventajas

- Es de libre distribución, cualquier persona puede regalarlo, venderlo o prestarlo.
- Es más seguro.
- Combate efectivamente la piratería de software.
- Menores tiempos de desarrollo debido a la amplia disponibilidad de herramientas y librerías.
- Se puede acceder a su código y aprender de él.
- Cualquier persona lo puede modificar, adaptándolo para realizar tareas específicas.
- Ahorro de dinero en la adquisición de licencias.
- Eliminación de barreras presupuestarias.
- Beneficio social y tecnológico para el país.

2.3.6.2 Desventajas

- No tiene garantía proveniente del autor.

- Se necesita dedicar recursos a la reparación de errores
- Las interfaces no son tan amigables como las del software propietario
- La mayoría de la configuración de hardware no es intuitiva
- El usuario debe tener nociones de programación
- La diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas con un mismo fin, etc., pueden crear confusión.

2.4. LINUX

2.4.1 ORÍGENES

Linux se origina en 1991, comenzó como un pasatiempo de su creador, Linus Torvalds y terminó siendo la razón de ayuda y cooperación entre numerosos desarrolladores de software de diferentes partes del mundo, aportando al mismo tiempo una importante pieza para la conformación de redes del mundo entero.

Torvalds al inicio dio a conocer su proyecto a la comunidad del internet, como si se tratara de un pasatiempo, nada tan grande ni comparable con lo que es ahora este Sistema Operativo, el mismo que era distribuido gratuitamente vía Internet y se ajustaba a las necesidades de quien lo requería, por lo que la Fundación para la Libertad del Software lo utilizó para el GNU, desde entonces Linux adoptó la política del Copyleft.

2.4.2 DEFINICIÓN

Linux es un Sistema Operativo basado en el Sistema Operativo Unix, que soporta gran número de usuarios y posibilita la ejecución de distintas tareas de forma simultánea.

2.4.3 CARACTERÍSTICAS DE LINUX

Entre las principales características de Linux tenemos que es:

- Multitarea
- Multiusuario
- Disponible para múltiples arquitecturas (Pc, Macintosh, etc)
- Multiprocesador
- Software libre
- Su costo de adquisición es mínimo
- El Kernel es mejorado de versión a versión
- Funciona en modo protegido 386 que consiste en algunos privilegios para acceder y modificar ciertas áreas de memoria, consta de 4 modos de protección: Ring0, Ring1, Ring2 y Ring3, los tres primeros poseen privilegios de administración y el Ring3 privilegios de usuario.

- Posee protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Posee una política de copia en escritura para compartir las páginas entre ejecutables
- Cuenta con memoria virtual usando paginación a disco
- Posee librerías compartidas de carga dinámica (DLL, Dynamic Link Library) y librerías estáticas.
- Emulación matemática en el núcleo
- Consolas virtuales múltiples
- Soporte para varios sistemas de archivo comunes
- Acceso transparente a particiones MS-DOS
- Posee un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- Cuenta con un sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- Permite la conectividad Appletalk Macintosh

- Software cliente y servidor Netware.
- Casi todos los programas están hechos en C.
- Las distribuciones de Linux más conocidas son: SLS, Debian, Lycoris, Gibraltar, Turbo Linux, Sorcerer, Mandrake, Red Hat, Corel Linux, Vector, Caldera Open Linux, etc.

2.4.4 COMPONENTES

2.4.4.1 Núcleo

Es el programa medular que ejecuta programas y gestiona dispositivos de hardware tales como los discos y las impresoras.

2.4.4.2 Shell

Proporciona una interfaz para el usuario. Recibe órdenes del usuario y las envía al núcleo para ser ejecutadas.

2.4.4.3 Sistema de archivos

Organiza la forma en que se almacenan los archivos en los dispositivos de almacenamiento. Los archivos están organizados en directorios. El núcleo, el shell y el sistema de archivos forman en conjunto la estructura básica del sistema operativo Linux. Con estos tres elementos puede ejecutar programas, gestionar archivos e interactuar con el sistema.

2.4.4.4 Las Utilidades

Son programas especializados, tales como editores, compiladores y programas de comunicaciones, que realizan operaciones de computación estándar. Linux contiene un gran número de utilidades. Algunas efectúan operaciones sencillas, otras son programas complejos con sus propios juegos de órdenes.

2.4.5 VENTAJAS Y DESVENTAJAS DE LINUX

2.4.5.1 Ventajas

- Posee todas las ventajas y características de Software Libre
- Soporta configuraciones de plataformas basadas en múltiples arquitecturas.
- Corrige los errores con gran rapidez

2.4.5.2 Desventajas

- La imagen errada de varias personas de que únicamente el software propietario es bueno y seguro.

2.4.6 EQUIVALENCIAS Y DIFERENCIAS ENTRE WINDOWS Y LINUX

(Anexo 2.1)

2.5. PHP (PROFESIONAL HOME PAGES)

2.5.1. INTRODUCCIÓN

Hasta hace poco para poder tener interactividad en el desarrollo de páginas web, se había aplicado tecnologías como: CGI, utilización de formularios, etc., pero actualmente existe el lenguaje PHP.

Con PHP se pretende conseguir la integración de las paginas HTML con aplicaciones que corran en el servidor como procesos integrados en el mismo, y no como un proceso separado; así como también que dichas aplicaciones sean totalmente independientes del navegador.

2.5.2. ORÍGENES

PHP fue creado por Rasmus Lerdorf a finales de 1994, pero la primera versión que se distribuyó fue a principios de 1995 y se llamó, Personal Home Page Tools (Herramientas para paginas web personales), consistía en un analizador sintáctico muy simple y una serie de utilidades comunes en las páginas web de ese entonces; el analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado PHP/FI version 2.

“PHP ha experimentado en poco tiempo una verdadera revolución y, a partir de sus funciones, en estos momentos se pueden realizar una multitud de tareas útiles para el desarrollo del web”.¹³

¹³ Fuente de consulta:<http://www.desarrolloweb.com/articulos/304.php?manual=12>

2.5.3. DEFINICIÓN

PHP es un lenguaje de programación soportado por HTML, orientado para la construcción de páginas web.

2.5.4. CARACTERÍSTICAS

Entre las principales características de PHP tenemos que:

- Es un procesador de hipertextos
- Su sintaxis está heredada de C, Java y Perl
- Permite crear páginas dinámicamente.
- Es software libre.
- Corre en múltiples plataformas y servidores web.
- Es independiente de la Base de Datos y del servidor Web.
- Permite tratar cadenas de texto o procesar ficheros.
- Permite crear documentos PDF (Formato Portátil para Documentos) o tratar con documentos XML (Xtensible Market Language)
- El código se ejecuta siempre en el servidor.
- Puede hacer lo que hace un script CGI.

- Soporte XML, WDDX (Web Distributed Data Exchange), XSLT(Extensible Style Language), DOM (Document Object Model)
- Soporta un único Safe Mode (mecanismo de seguridad).
- Tratamiento de cookies (archivos de texto empleados por los servidores Web para almacenar y recuperar información de sus visitantes).
- Accesos restringidos
- Comercio electrónico
- Posee funciones de propósito general como: funciones matemáticas, explotación de cadenas, fechas, corrección ortográfica, etc.

2.5.5. TAREAS PRINCIPALES DEL PHP

2.5.5.1. Funciones de Correo Electrónico

Envío de un e-mail a una persona o lista de personas con los campos e-mail de procedencia, asunto, persona a responder.

2.5.5.2. Gestión de Bases de Datos

El lenguaje PHP ofrece interfaces para administración y manejo de diferentes bases de datos como por ejemplo: MySql y Oracle que utilizan el modelo relacional como sistema de gestión de Bases de datos (Ver MySql, Modelo relacional).

2.5.5.3. Gestión de archivos

Con PHP se puede crear, borrar, mover, modificar cualquier tipo de operación razonable para la gestión de archivos. Además se puede transferir archivos por FTP (File Transfer Protocol) a partir de sentencias.

2.5.5.4. Tratamiento de imágenes

PHP permite uniformar en tamaño y formato miles de imágenes recibidas día tras día, todo esto gracias a su librería de funciones gráficas.

2.6. MYSQL

2.6.1. INTRODUCCIÓN

Hoy en día para poder almacenar los datos en Internet contamos con varias bases de datos, una de ellas es MySQL, que se utiliza para darle dinamismo a una web en PHP, ofreciendo una variedad de opciones. Esta base de datos usa tablas para almacenar la información.

2.6.2. ORÍGENES

MySQL empezó como un proyecto de programación de Michael Widenis en la década de los 90, al crear su propio administrador de datos para la aplicación que estaba desarrollando. Hasta ese momento usaba msql, pero se dio cuenta que lo podía mejorar creando el MySQL.

Posteriormente se creó la empresa sueca, llamada Tux dedicada a crear nuevas versiones de MySQL.

2.6.3. DEFINICIÓN

MySQL es un administrador de bases de datos relacional SQL (Structured Query Language),

2.6.3.1. Modelo Relacional

El modelo relacional se basa en el concepto matemático denominado relación que gráficamente se puede representar como una tabla, cada relación matemática, posee un nombre que es único.

En el modelo relacional la base de datos a nivel lógico es percibida por el usuario como un conjunto de tablas

Tabla.- Es el conjunto de dos o más registros (un registro está formado por varios campos)

2.6.4. CARACTERÍSTICAS

Entre las principales características de MySQL tenemos que:

- Es software libre.
- Posee arquitectura cliente-servidor (Un servidor es un computador remoto ubicado en algún lugar de la red que proporciona información según petición. Un cliente funciona en su computador local, se comunica con el servidor remoto, y pide a éste información, quien envía la información solicitada).
- Se Integra con cualquier lenguaje de programación.

- Permite definir reglas o restricciones que garanticen la integridad referencial de los registros.
- Por estar en constante desarrollo, MySQL Server ofrece una variedad de funciones.

CAPITULO III

WEB SERVICES

3.1. INTRODUCCIÓN

El World Wide Web ha pasado de ser un medio para la publicación de información y contenidos, a convertirse en una plataforma para el diseño y desarrollo de aplicaciones informáticas distribuidas. Con el surgimiento de los Web Services o Servicios Web nace una nueva generación de aplicaciones distribuidas. Para empezar a tratar este tema, a continuación se realiza un pequeño recuento desde los comienzos de las aplicaciones distribuidas hasta los Web Services.

La aparición de las redes y sobretodo de Internet, ha permitido una revolución en la informática permitiendo el intercambio masivo de información, la computación distribuida, la videoconferencia, y otras aplicaciones ya se han establecido rápidamente en nuestra sociedad. Hoy en día, resulta difícil imaginarse la informática sin la conexión y comunicación entre computadores.

La informática ha evolucionado hacia la computación distribuida basada en redes de computadores. Al hablar de redes no únicamente se refiere al Internet que es la red de redes, sino a pequeñas redes empresariales o incluso domésticas, que permiten el intercambio de información entre usuarios.

Las aplicaciones distribuidas basadas en la intercomunicación de procesos o máquinas a través de la red son cada vez más importantes debido al enorme incremento de posibilidades respecto a las aplicaciones tradicionales, permitiendo la

interconexión entre los diferentes sistemas operativos y aplicaciones a través de la red. Hoy en día las organizaciones desarrollan sus aplicaciones distribuidas basadas en tecnologías, como COM (Component Object Model, Modelo de Componentes de Objeto), CORBA (Common Object Request Broker Architecture, Arquitectura de requerimiento de Intermediación de Objetos Comunes), EJB (Enterprise Java Beans), Java/RMI (Java Remote Method Invocation, Invocación a métodos remotos de Java) entre otros. Cada una proporciona un marco de trabajo para el acceso a objetos remotos, mediante la solicitud a un servidor pero poseen ciertas limitantes que las hacen poco viables, entre estas se puede citar la falta de interoperabilidad y la dependencia a la arquitectura de trabajo, un claro ejemplo es COM que está muy ligado a Windows, mientras que CORBA tiene muchas implementaciones de diversos fabricantes.

Para suplir estos inconvenientes surgen los Web Services que al estar basados en estándares como XML (eXtensible Markup Language, Lenguaje de Marcas extensible) permiten que las aplicaciones se integren y compartan información independientemente de cómo hayan sido creadas, cuál sea el sistema operativo o la plataforma en que se ejecutan. Lo que se desea es dejar de concebir al Internet como un camino de una sola dirección: la interacción comienza y termina en dos puntos claramente definidos: la petición del usuario y la respuesta de la aplicación informática, en donde por lo general se tiene un browser o navegador mediante el cual se accede a una empresa "X" y a la cual se conecta para obtener servicios, y si se necesita acceder a la empresa "Y" de igual forma se procede a conectarse para acceder a los servicios de esta otra empresa, y eso es lo que se hace por cada empresa a la que desee acceder. Al tener un Internet de dos vías se puede mandar y recibir información de forma proactiva, donde las aplicaciones de las empresas puedan interactuar entre sí y devolver la información que realmente se necesita.

Para explicación, se pone en manifiesto lo que el presente trabajo tiene como objetivo realizar:

La mayoría de las Universidades y Escuelas Politécnicas del país tienen desarrolladas aplicaciones web a las cuales puede acceder cualquier usuario. Lo que se tienen es a una persona conectándose y haciendo transacciones o búsquedas en cada una de estas aplicaciones. El presente trabajo evitará al usuario conectarse a distintos sitios ya que realizará un compendio de la información acerca de las ofertas académicas que brindan las Universidades y Escuelas Politécnicas del Centro de País. Para la demostración se utilizará la información de la oferta académica de la Escuela Politécnica del Ejército sede Latacunga (ESPE-L), y de la Universidad Técnica de Cotopaxi (UTC). Esta información podrá ser visualizada por los usuarios desde cualquier navegador y desde cualquier aplicación que haga uso del Web Services que se encuentra en el sistema OFACUEP (Oferta Académica de las Universidades y Escuelas Politécnicas).

3.2. DEFINICIONES DE WEB SERVICES.

Según la W3C World Wide Web Consortium: “Web Service es un sistema de software, cuyas interfases y enlaces (bindings) son definidos y representados usando XML. Este sistema puede ser descubierto y usado por otros sistemas de software. Esa interacción debe darse según la manera representada en la definición, usando mensajes de acuerdo a protocolos de Internet.”

Según Graham Glass: “Web Services es una función o colección de funciones que están empaquetadas en una entidad simple y ofrecida en la Red para que otros programas la usen. Son building blocks para crear sistemas distribuidos y permitir que empresas y usuarios en general, puedan lograr de una forma rápida y barata, su inserción en la Web” ¹⁴

¹⁴ Fuente de consulta

http://www.fcad.uner.edu.ar/Extension/2_jor_adm_inf/Ponencias/Molinari/Arquitecturas.ppt

3.3. WEB SERVICES.

Los Web Services o también llamados Servicios basados en tecnología Web han comenzado a ser un tema de interés para el mundo de los negocios y de las comunicaciones.

En los últimos dos años se han ido construyendo, una serie de elementos que permitirán que la Internet se introduzca de manera más dramática en la forma en que nos comunicamos, compramos y nos relacionamos. La misma que se trata de la creación de aplicaciones informáticas que, a diferencia del Web, no necesariamente tienen una interfaz, sino que funcionan internamente en los sistemas, facilitando el intercambio de datos entre diferentes programas que de otra manera, no podrían relacionarse entre ellos.

Un Web Service se suele definir como una unidad o un módulo de aplicación capaz de ofrecer datos o servicios de procesamiento a otras aplicaciones informáticas. De esta manera una aplicación puede ofrecer distintos servicios a otras aplicaciones. Las características de estos servicios son que se solicitan a través del web y que los resultados de su ejecución también se devuelven a la aplicación solicitante a través del web.

Un Servicio Web o Web Services consiste en una función disponible en un servidor conectado al Internet. Esta función puede consistir en cualquier cosa como realizar un simple cálculo con unos datos que se le envían como parámetro, acceder a una base de datos para recuperar un conjunto de registros, o validar datos, etc.

El Servicio Web podrá ser solicitado desde otro programa informático que se ejecute en un computador conectado al Internet. Junto a la solicitud de la ejecución, se pueden enviar al computador que ofrece el servicio los parámetros que el Web Services remoto tomará como base para el cálculo o la función.

La aplicación que actúa como cliente debe conocer:

La URL del servidor remoto que ofrece el servicio,

El nombre del servicio que se solicita, y

Los parámetros que se deben enviar junto con la llamada al servicio.

Estos datos se enviarán mediante HTTP (HiperText Transference Protocol, Protocolo de Transferencia de Hipertexto), el servidor que ofrece el servicio web leerá los parámetros que se le han enviado, llamará a un componente o programa encargado de implementar el servicio, y los resultados que se obtengan de su ejecución serán devueltos al servidor que solicitó la ejecución del servicio.

Las versiones más innovadoras de los Web Services contemplan la posibilidad de que dos servidores entre los que no ha habido ninguna transacción y que ni siquiera poseen acuerdos previos para interconectarse, puedan hacerlo. Toda esta actividad entre servidores puede acompañarse con interfaces Web, las cuales pueden servir para realizar desde un simple monitoreo de los procesos, hasta controlarlos completamente, utilizando siempre un browser o navegador.

Los Web Services no actúan solos, éstos están basados en ciertos estándares y protocolos que posibilitan la comunicación entre aplicaciones.

Servicio de Transporte: Esta capa es la responsable de transportar los mensajes entre las aplicaciones, y pueden ser bajo HTTP, SMTP (Simple Mail Transference Protocol, Protocolo simple de Transferencia de Correo) o FTP (File Transference Protocol, Protocolo de Transferencias de Archivos).

Mensajería XML: Esta capa es la responsable de codificar los mensajes en el formato de XML para que puedan ser entendidos en ambos extremos de la comunicación, esto se da a través XML-RPC o SOAP (Simple Object Access

Protocol, Protocolo de Acceso Simple a Objetos). La Descripción del Servicio: Esta capa es responsable de describir la interfaz de un Web Services. La descripción de servicio actualmente, se realiza vía WSDL (Web Services Description Language, Lenguaje de Descripción de Servicios Web). El descubrimiento del Servicio : Esta capa es la responsable de centralizar los servicios en un registro común, y proveer la funcionalidad de encontrar y publicar los Web Services. Esta función está a cargo de UDDI (Universal Discovery Description and Integration; Integración, Descripción y Descubrimientos Universal).

Estas capas serán expuestas con mayor amplitud en páginas posteriores.

3.3.1. XML (EXTENSIBLE MARKUP LANGUAGE, LENGUAJE DE MARCAS EXTENSIBLE XML)

“XML, es una especificación que permite por medio de etiquetas inventadas clasificar la información para su posterior uso en páginas Web, bases de datos y otras aplicaciones. Separar los datos de una página web para apoyar la creciente necesidad de comunicar HTML con base de datos permitió la aparición de XML.”¹⁵

Se dice que el Web revolucionó la comunicación entre humanos y aplicaciones pero es XML el que está revolucionando la comunicación entre aplicaciones o, de forma más general, la comunicación entre computadores, pues ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información, evitando que ésta quede ligada o conserve las características de determinado programa permitiendo la interoperabilidad entre equipos. Día a día aparecen nuevas propuestas para estandarizar formatos basados en XML. Por esto el XML se está convertido en una tendencia y se está consolidando. Actualmente los programadores están realizando sus programas basados en XML, de esta manera si un día

¹⁵ Fuente de Consulta: Revista PCWORLD abril del 2004 artículo “ XML: Estructurando los datos”

necesitan migrar sus datos a otras aplicaciones, les resultará más sencillo y no necesitarán volver a programar toda la aplicación.

Para comprender de mejor manera a XML, primero se explica en qué consiste un lenguaje de marcas.

3.3.2. LENGUAJE DE MARCAS

Las marcas son códigos que indican a un programa cómo debe tratar su contenido, por ejemplo, si se desea que un texto aparezca con un formato determinado, dicho texto debe ir delimitado por la correspondiente marca que indique como debe ser mostrado en pantalla o impreso.

“IBM creó las marcas como texto accesible desde cualquier programa que maneje texto plano basado en código ASCII, a esta norma se la denominó GML (General Markup Language). Más tarde GML pasó a manos de ISO y se convirtió en SGML (ISO 8879), Standart Generalized Markup Language. Esta norma es la que se aplica desde entonces a todos los lenguajes de marcas, cuyo ejemplo más conocido es el HTML.”¹⁶

Los lenguajes de marcas son sistemas complejos de descripción de información, normalmente documentos, que si se ajustan a SGML, se pueden controlar desde cualquier editor ASCII: NotePad, Word, etc.

“Las etiquetas describen la estructura de un documento a la vez que identifican su contenido”¹⁷. Suelen describirse por textos encerrados entre signos de "menor" (<) y "mayor" (>), siendo lo más usual que existan una marca de principio y otra de final. Se puede decir que existen tres utilizaciones básicas de los lenguajes de marcas: los

¹⁶ Fuente de Consulta: <http://www.desarrolloweb.com/articulos/450.php?manual=18>

¹⁷ Tomado del Libro: XML Manual de Referencia

que sirven principalmente para describir su contenido como las aplicaciones de bases de datos, los que sirven más que nada para definir su formato por ejemplo los programas de tratamiento de textos y los que realizan las dos funciones indistintamente así como el HTML que es la muestra más conocida de este modelo.

“El marcado de XML es un metalenguaje, lo que significa que se puede utilizar para describir otros lenguajes. Esto permite crear marcas de XML propias que proporcionen información acerca del contenido para el que se está creando la estructura del documento basada en XML”¹⁸

El XML se puede usar para infinidad de trabajos y aporta muchas ventajas en amplios escenarios. A continuación algunas ventajas del XML en algunos campos prácticos:

Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.

Migración de datos. Si tenemos que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.

Aplicaciones Distribuidas. Hasta ahora cada browser o navegador interpreta la información a su manera y los programadores del web tienen que hacer unas cosas u otras en función del browser del usuario.

Un documento XML puede contener muchos tipos de información. Es decir, pueden haber muchos lenguajes escritos en XML para cualquier tipo de usuarios. Por ejemplo, si lo utiliza el usuario Médicos podría crear un lenguaje en XML específico

¹⁸ Tomado del Libro: XML: Manual de Referencia

para almacenar diagnósticos de los pacientes y llamarse XMPacientes. Si las tiendas de alquiler de películas utilizan XML podrán crear sus propios lenguajes XMPeliculas para guardar la información de las películas, etc.

Por esta razón se pueden crear infinitos lenguajes a partir del XML. Para especificar cada uno de los usos de XML, o lo que es lo mismo, para especificar cada uno de los sublenguajes que se puede crear a partir de XML. En estos casos, es necesario validar que un documento XML es correcto, es decir, que las etiquetas que se usan son correctas y que están anidadas de la forma adecuada. Para ello se pueden usar dos formalismos: DTD (Data Type Dictionary), o bien XML Schema.

El DTD, Definition Type Document o Definición de Tipo de Documentos en español, es generalmente un archivo (o varios usados conjuntamente) que contiene la definición formal de un documento XML en particular. El mismo que define los nombres que pueden utilizarse en los elementos, dónde pueden aparecer y cómo se interrelacionan entre ellos, tiene una sintaxis especial, distinta de la de XML por lo que se encontró una manera de escribir en XML la definición de otro lenguaje XML. Se definió entonces el lenguaje XML Schema, aunque resulta un poco más complicado que especificarlo en DTD.

SGML utiliza Definiciones de Tipo de Documentos(DTD) para describir los elementos o etiquetas disponibles en un tipo de documento específico. Sin embargo, el diseño y construcción de una DTD puede ser una tarea compleja, por lo que XML ha sido diseñado para ser usado con o sin una DTD. Sin DTD significa que se puede inventar etiquetas sin tener que definirlos formalmente. Para que esto funcione, un archivo sin DTD define sus propias etiquetas, informalmente. Pero cuando una aplicación XML, como un visualizador encuentra un archivo sin DTD, necesita ser capaz de entender la estructura del documento que está leyendo, y dado que no tiene DTD que le informe al respecto, se han hecho algunas modificaciones en las reglas.

Por ejemplo, el elemento de HTML se define como vacío: no tiene etiqueta de fin. Sin una DTD, una aplicación XML no tendría manera de conocer si debe esperar una etiqueta de fin para un determinado elemento, luego ha tenido que introducirse el concepto de XML 'bien-formado'.

Todos los documentos XML deben estar 'bien-formados':

si no se utiliza DTD, el documento debe comenzar con una Declaración de Documento único (SDD, Standalone Document Declaration) que indique:

```
<?XML version="1.0" standalone="yes"?>
```

Todas las etiquetas deben estar balanceadas: esto es, todos los elementos que contengan datos de tipo carácter deben tener etiquetas de principio y fin.

Todos los valores de los atributos deben ir entrecomillados.

Cualquier elemento vacío por ejemplo aquellos que no tienen etiqueta final como , <HR>, y
 y otros de HTML deben terminar con '>' o se los deben hacer no vacíos añadiéndoles una etiqueta de fin.

Un dato muy importante acerca de los DTD o XML Schema es que estos lenguajes también permiten comprobar la integridad de los datos.

Ejemplo de un documento XML de la cuenta bancaria de un cliente de MiBanco

```
<?xml version="1.0"?>
<!DOCTYPE CLIENTE [
<!ELEMENT CLIENTE(NOMBRE, DIRECCION,TELEFONO)>
    <!ELEMENT NOMBRE (#PCDATA)>
    <!ELEMENT DIRECCION (#PCDATA)>
    <!ELEMENT TELEFONO (#PCDATA)>
]>
```

```
<CLIENTE>
<NOMBRE>"Luis Perez" </NOMBRE>
<DIRECCION> "Alpallana 555 y Amazonas" </DIRECCION>
<TELEFONO> "022598745" </TELEFONO>
<CUENTA xmlns="http://www.mibanco.com/banco"> 729-1269-4785
</CUENTA>
</CLIENTE>
```

3.4. SOAP (SIMPLE OBJECT ACCESS PROTOCOL, PROTOCOLO DE ACCESO SIMPLE A OBJETOS)

SOAP es un protocolo relativamente nuevo. Define una manera de dar formato a un código XML para usarlo como mensaje. Los Servicios Web utilizan SOAP a manera de codificación de mensajes para comunicar la petición y la respuesta entre los dos programas. En su especificación, define un conjunto de reglas sobre como armar los mensajes y cómo ser utilizados para funcionar como un mecanismo de invocación de procedimientos remotos similar a RPC (Llamada a procedimientos remotos).

SOAP es un protocolo basado en XML con su propio esquema que permite comunicar componentes y aplicaciones mediante protocolos normales como HTTP, SMTP o TCP/IP que posibilita hacer llamadas a procedimientos remotos usando XML como lenguaje común y HTTP como protocolo de transporte.

“Actualmente las organizaciones han adoptado la estrategia de desarrollo de aplicaciones distribuidas usando tecnologías diversas, como COM (Component Object Model), CORBA (Common Object Request Broker Architecture), EJB (Enterprise Java Beans). Una propuesta reciente es SOAP (Simple Object Access Protocol), que propone resolver los problemas de falta de interoperabilidad entre las opciones anteriores, tomando como base protocolos ya establecidos y con gran

aceptación en Internet, como HTML y XML. Se pretende mostrar una visión del funcionamiento de SOAP para la construcción de servicios Web. “¹⁹

La especificación SOAP define dos modelos de mensajes:

Un mensaje que se enviará desde la aplicación cliente a la aplicación servidor, solicitando la ejecución de un método al que se pasan una serie de parámetros.

Y un mensaje que se enviará desde la aplicación servidor a la cliente, y que contendrá datos XML con los resultados de la ejecución del método solicitado.

Estructura de un Mensaje SOAP

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<soap:Envelope xmlns:soap= "http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:header>  
  marca opcional  
</soap:header>
```

```
<soap:body> lugar donde van los  
mensajes </soap:body>
```

```
</soap:Envelope>
```

Ejemplo: mensaje para acceder a una cuenta en Mibanco

```
<soap:Envelope xmlns:soap= "http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Body>
```

```
<GetSaldo
```

¹⁹ Fuente de Consulta: http://www.xml.com.ve/noticias_1.htm

```
xmlns="http://www.qwickbank.com/bank"> <CUENTA>729-1269-4785
</CUENTA>
</GetSaldo>
</soap:Body>
</soap:Envelope>
```

Ejemplo: mensaje que retorna el saldo de la cuenta (balance) en QuickBank

```
<soap:Envelope xmlns:soap= "http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<GetSaldoResponse xmlns="http://www.qwickbank.com/bank">
<Saldo>3,822.55</Saldo>
</GetSaldoResponse>
</soap:Body>
</soap:Envelope>
```

3.5. WSDL (WEB SERVICES DESCRIPTION LANGUAGE, LENGUAJE DE DESCRIPCIÓN DE SERVICIOS WEB)

Un archivo WSDL es un documento XML que describe un conjunto de mensajes SOAP y cómo se realiza el intercambio de mensajes. Como WSDL está basado en XML, es legible y editable pero en la mayoría de casos, se genera y se consume por parte de software.

WSDL es la gramática que define los distintos métodos o funciones que están disponibles en un servicio Web así como el número o tipos de argumentos que se le deben pasar y de igual forma el tipo de dato que devolverá la función como resultado.

Aunque SOAP define el formato del mensaje, también necesitará una forma para describir la gramática de los mensajes. Esto permite a los clientes y servidores

generar automáticamente y reconocer mensajes SOAP satisfactorios. El Lenguaje de descripción del Servicio Web WSDL es una forma de describir a los Servicios Web. Define los métodos admitidos por un Servicio Web, y los parámetros de cada mensaje. En muchas formas, WSDL es el contrato que vincula al cliente con el servidor en donde se especifica los servicios que el servidor se compromete a ofrecer al cliente, siempre que este le solicite de la manera adecuada.

La notación que utiliza un archivo WSDL para describir formatos de mensajes está basada en el estándar XML lo cual significa que es neutral respecto del lenguaje de programación y que está basado en estándares, lo que lo hace apropiado para describir interfaces de servicios Web accesibles desde una amplia variedad de plataformas y lenguajes de programación. Además de describir el contenido de los mensajes, WSDL define dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. Esto significa que el archivo WSDL define todo lo necesario para escribir un programa que interactúe con un servicio Web

3.5.1 Anatomía de un documento WSDL

Si se examina cada parte de un documento WSDL se encontrará:

<definitions>. El elemento <definitions> contiene la definición de uno o más servicios. En la mayoría de los casos, un archivo WSDL define un servicio únicamente. Seguido de la etiqueta de definición se encontrarán declaraciones de algunos atributos.

name: Este atributo es opcional que indica el propósito del servicio.

targetNamespace: Este atributo define un namespace lógico para la información sobre el servicio que debe ser único.

xmlns:soap and xmlns:xsd: Estos son namespace estandarizados para la definición de un documento WSDL.

xmlns: el namespace de un documento WSDL es <http://schemas.xmlsoap.org/wsdl/>. Y todos los Tags como `<definitions>`, `<message>`, and `<service>` residen en este namespace

Dentro de la etiqueta `<definitions>` se encuentran tres secciones conceptuales:

<message> y **<portType>**, describe qué operaciones provee el servicio.

<binding>, describe cómo se invocan las operaciones.

<service>, describe dónde se ubica el servicio.

<documentation>, cualquier elemento WSDL puede contener información del servicio para el usuario.

Como la creación de un documento WSDL resulta compleja, las distintas implementaciones de SOAP permiten generar estos archivos de forma sencilla, sin necesidad de conocer los elementos y la estructura del esquema XML en el que se basan, este es el caso de la librería de PHP nuSoap que permite generar automáticamente documentos WSDL.

3.6. UDDI (UNIVERSAL DISCOVERY DESCRIPTION AND INTEGRATION , INTEGRACIÓN, DESCRIPCIÓN Y DESCUBRIMIENTO UNIVERSAL)

“UDDI (Universal Description, Discovery, and Integration), que se encuentra en desarrollo, para describir, publicar, y localizar servicios Web que provee alguna empresa o compañía. En pocas palabras, el UDDI es una especificación para un registro distribuido de información sobre servicios Web.”²⁰

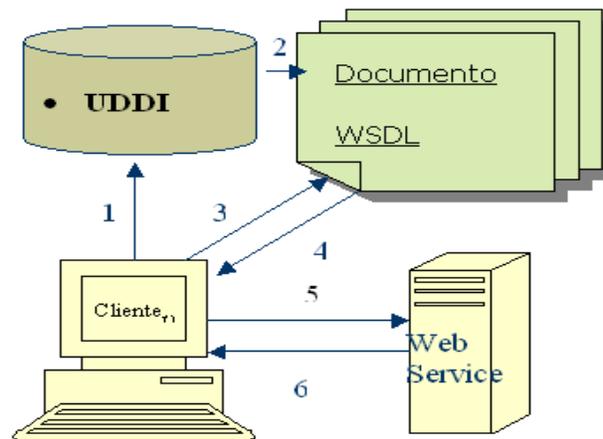
Siempre ha sido un reto la comunicación entre los negocios a nivel de aplicaciones, dada la vasta existencia de plataformas, herramientas, mecanismos, y procesos que cada quien utiliza. La popularidad reciente del XML (eXtensible Markup Language) en estos sitios, promete una solución para el intercambio de datos de una forma transparente. También, la evolución de protocolos como el SOAP (Simple Object Access Protocol) proporciona una plataforma para el intercambio de servicios sobre la red.

Entonces, si el mecanismo de comunicación entre las plataformas es en XML, y la forma de comunicación es en SOAP, ¿cómo sabrán con quien comunicarse y dónde encontrar otros negocios?. La respuesta es el UDDI.

En sí UDDI viene a ser como las páginas amarillas de los Servicios Web. Al igual que con las páginas amarillas tradicionales, se puede buscar una empresa que ofrezca los servicios que necesitamos, obtener información sobre el servicio ofrecido y contactar con alguien para más información. Naturalmente, también se puede ofrecer un Servicio Web sin registrarlo en UDDI, al igual que se puede abrir un negocio y confiar en la publicidad boca a boca, pero si se desea llegar a un mercado significativo, se necesita UDDI para que los clientes puedan encontrar el servicio.

²⁰ Fuente de Consulta: http://www.fisica.uson.mx/carlos/WebServices/WS_UDDI.htm

El directorio UDDI también incluye varias formas de buscar los servicios que necesitamos para construir nuestras aplicaciones. Por ejemplo, podemos buscar los proveedores de un servicio en una ubicación geográfica específica o un negocio de un tipo específico. El directorio UDDI proporcionará información, contactos, enlaces e información técnica para permitirnos evaluar qué servicios satisfacen nuestros requerimientos.



Arquitectura de los Web Services Figura 3. 1

1. El cliente pregunta al UDDI para ubicar un servicio.
2. UDDI le indica al cliente un documento WSDL.
3. El cliente accede al documento WSDL.
4. WSDL provee lo necesario para interactuar con el Web Service.
5. El cliente envía un requerimiento usando SOAP.
6. Web Service retorna una respuesta SOAP

En esta arquitectura hay tres roles bien definidos:

- Proveedor del servicio
- Lugar o guía para la búsqueda de servicios o Agente
- Solicitante del servicio o Consumidor



Roles de los Web Services Figura 3.2

3.7. VENTAJAS Y DESVENTAJAS DE LOS WEB SERVICES.

3.7.1. VENTAJAS

- Una de las ventajas de los web services es que al usar protocolos como HTTP puede traspasar un Firewall.
- Los protocolos, que se usan en los web services, se basan en XML.
- Al ser XML universalmente aceptado para transmitir datos en sistemas distribuidos, lo vuelve atractivo para los programadores.

- Otra de las ventajas para los web services es que la mayoría de los sistemas operativos vienen con un parser XML que lee el documento XML y lo expone al programa cliente.
- Los web services poseen una ventaja sobre RPC ya que el puerto 80 siempre está abierto y el 135 para RPC se inhabilita por potenciales vulnerabilidades.
- Permiten hacer un mejor uso de las facilidades que ofrece Internet en cuanto a programación distribuida.
- Ofrecen soluciones técnicas no solamente al área de la informática, también al marketing y áreas comerciales.

3.7.2. DESVENTAJAS

- Los Web Services están en una fase inicial, por lo tanto no son muy conocidos ni explotados.
- Los Web Services no son exentos de las fallas a los que están expuestos cualquier sistema distribuido.
- El protocolo SOAP resulta más complejos de comprender que los IDL (Interface Definition Language, Lenguaje de Definición de Interfaz) de CORBA.
- El puerto 80 HTTP en un momento dado puede llegar a saturarse.

CAPÍTULO IV

DESARROLLO DE LA APLICACIÓN OFERTA ACADÉMICA DE LAS UNIVERSIDADES DEL CENTRO DEL PAÍS -OFACUEP

En este capítulo se muestra el desarrollo de la Aplicación Oferta Académica de las Universidades del Centro del País.

4.1 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE

4.1.1 INTRODUCCIÓN

Este documento es la Especificación de Requisitos de Software (ERS) del sistema OFACUEP, y es a la vez el resultado de la observación a los estudiantes (usuarios) que desean continuar sus estudios superiores.

Esta especificación se ha basado en el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998”.

4.1.2 PROPÓSITO

El objeto de la Especificación de Requisitos es definir de manera clara y concisa las funcionalidades y restricciones del sistema OFACUEP. El presente documento va

dirigido al personal de desarrollo, a las personas que lo pondrán a prueba y a los usuarios finales.

Esta especificación está sujeta a revisiones y modificaciones por parte de las partes implicadas, de acuerdo a sus necesidades, hasta lograr la aprobación y agrado de todas las partes.

Una vez revisado y aprobado será la base para la construcción del sistema.

4.1.3 ÁMBITO DEL SISTEMA

La futura aplicación recibirá el nombre de “OFACUEP”, abreviatura de Oferta Académica de las Universidades del Centro del País, la misma que permitirá la búsqueda de información de las Universidades y Escuelas Politécnicas para las personas que deseen continuar sus estudios superiores, con el fin de facilitar a los interesados la toma de decisiones. En sí, el objetivo del sistema es ser un aporte para que las próximas generaciones de estudiantes que aspiran continuar sus estudios superiores, no se desalienten en la búsqueda de la universidad más idónea para ellos, por lo que queremos poner en sus manos una herramienta en donde las Universidades y Escuelas Politécnicas conjuguen la información de su Oferta Académica y se sirvan de la tecnología para que los estudiantes tomen una mejor elección de acuerdo a la Universidad, especialidad, título, modalidad, ciudad, nivel, duración, escogiendo la que mejor se acople a sus necesidades.

OFACUEP se encargará de brindar un espacio a las Universidades y Escuelas Politécnicas para que publiquen su oferta académica, en donde el usuario podrá hallar información acerca del nombre de la Universidad o Escuela Politécnica, especialidades, título a obtener, modalidad, ciudad, nivel de estudio, duración e infraestructura

OFACUEP partirá teniendo en cuenta que no existe ninguna base informática sobre la Oferta Académica de las Universidades que de alguna u otra manera facilite su automatización.

OFACUEP será lanzado como prueba piloto tomando en cuenta únicamente dos Universidades como son: La Escuela Superior Politécnica del Ejército Sede Latacunga y La Universidad Técnica de Cotopaxi, ya que dos son suficientes para demostrar la funcionalidad de esta investigación, y posteriormente implementarla en las universidades que se desee.

4.1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

A continuación se definen las definiciones, acrónimos y abreviaturas que utilizaremos en el ERS.

4.1.4.1 Definiciones

Usuario	Cualquier persona que tiene la necesidad de informarse sobre la oferta académica de las Universidades y Escuelas Politécnicas.
Universidad	Universidad o Escuela Politécnica que desea publicar su Oferta Académica en OFACUEP.
Administrador Universitario	Persona encargada de manejar los datos de la Universidad.
Administrador Central	Persona encargada de administrar las cuentas y verificar los datos de las Universidades
Oferta Académica	Ofrecimiento de las distintas carreras y/o especialidades de las Universidades.
Buscador de Carreras	Funcionalidad de la aplicación que permite al usuario realizar búsquedas
Interfaz Universidad	Interfaz que permite a las Universidades usar la información almacenada en el repositorio de OFACUEP.

Sistema Universidad	Sistema de la Universidad o Escuela Politécnica que solicita y utiliza el servicio de OFACUEP.
------------------------	--

Tabla 4. 1

4.1.4.2 Acrónimos

ERS	Especificación de Requisitos de Software
-----	--

Tabla 4. 2

4.1.4.3 Abreviaturas

OFACUEP	Oferta Académica de las Universidades y Escuelas Politécnicas.
CONESUP	Consejo Nacional de Educación Superior

Tabla 4. 3

4.1.5 REFERENCIAS

IEEE Recommended Practice for Software Requeriments Specification. ANSI/IEEE std 830, 1998.

4.1.5.1 Visión General del Documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS del sistema OFACUEP. En la sección 4.2 se da una descripción general, con el fin de conocer las principales funciones que debe realizar OFACUEP, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en mayor detalle.

En la sección 4.3 se definen detalladamente los requisitos que debe satisfacer OFACUEP.

4.1.6 DESCRIPCIÓN GENERAL

Esta sección nos permite conocer de una manera general las funciones principales que el sistema debe realizar.

4.1.6.1 Perspectiva del producto

El sistema OFACUEP interactuará con las Universidades y los Usuarios, al poseer un compendio de las Universidades que a él se suscriban, y brindando a los usuarios una herramienta en la toma de decisiones para realizar sus estudios superiores, evitando la molestia de que los usuarios ingresen por separado a cada sitio web de las Universidades.

4.2 FUNCIONES DEL SISTEMA

En términos generales, OFACUEP deberá proporcionar soporte a las siguientes tareas:

4.2.1 GESTIÓN ADMINISTRADOR DENTRAL

El Administrador Central debe darse de alta en OFACUEP para poder interactuar con el Administrador Universitario.

El Administrador Central es la persona encargada de verificar que los datos de las Universidades sean verídicos, cabe indicar que este proceso será realizado de forma

manual. En el caso de que los datos de la Universidad o Escuela Politécnica sean erróneos el Administrador Central procederá a comunicar al Administrador Universitario el inconveniente.

4.2.2 GESTIÓN UNIVERSIDAD

El Administrador Universitario a través del sitio web de OFACUEP suscribirá a la Universidad a la que representa, para que posteriormente a través de un Login y un Password proporcionado por el Administrador Central, ingrese y maneje los datos de dicha Universidad en el sistema y puedan ser publicados en OFACUEP.

4.2.3 BÚSQUEDA DE ESPECIALIDADES

El usuario podrá realizar búsquedas acerca de las especialidades, accediendo a OFACUEP desde su navegador, pudiendo ingresar los datos sobre lo que desea averiguar, estos datos pueden combinarse entre:

- Nombre de la Universidad
- Nombre de la Especialidad o Carrera
- Título que se Oferta
- Modalidad de estudio(Presencial, Semipresencial o a Distancia)
- Ciudad
- Nivel (Pregrado o Postgrado)

4.2.4 INTERFAZ UNIVERSIDAD

La interfaz Universidad realizará las búsquedas solicitadas por el usuario o el Sistema Universidad detalladas en la función 4.2.3 (ver Búsqueda de Especialidades)

4.2.5 CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios deberán tener conocimiento de computación y manejo de internet.

4.2.6 RESTRICCIONES

A continuación se especifica las funciones que no serán implementadas en OFACUEP.

4.2.6.1 Requisitos no Funcionales

OFACUEP, no se encargará de realizar o receptor ningún tipo de inscripciones o matrículas de estudiantes, así como tampoco especificará el mapa curricular de las diferentes carreras ni el costo de la matrícula en las Universidades.

4.2.7 SUPOSICIONES Y DEPENDENCIAS

4.2.7.1 Suposiciones

Se asume que los datos de la Universidad ingresados por el administrador universitario son exactos y verdaderos, de tal manera que los usuarios obtengan datos confiables, por lo tanto la responsabilidad sobre la credibilidad de la información ingresada al sistema recae en la Universidad al delegar a una persona confiable que ingrese sus datos como lo es el Administrador Universitario.

4.2.7.2 Dependencias

OFACUEP funcionará autónomamente sin la necesidad de otros sistemas, sin embargo los sistemas externos (sistema Universidad) que deseen obtener el servicio que brinda OFACUEP, si requieren de éste.

4.3 REQUISITOS ESPECÍFICOS

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por OFACUEP. Todos los requisitos aquí expuestos son esenciales, es decir, OFACUEP no sería aceptable sin alguno de los requisitos aquí presentados.

4.3.1 GESTIÓN ADMINISTRADOR CENTRAL

Req(01) El administrador central deberá darse de alta en OFACUEP ingresando los siguientes datos: Nombre, Apellido, Número de cédula, email, dirección, teléfono, login y password.

Req(02) Si los datos del administrador central se insertaron correctamente, OFACUEP guardará los datos y presentará un mensaje de aviso notificándolo, si no se insertaron correctamente, OFACUEP no guardará los datos y presentará un mensaje de aviso notificándolo.

Req(03) Una vez cumplido con el Req(02) OFACUEP permitirá al administrador central autenticarse a través del login y password, se deberá verificar que el login y

password sean correctos, si es así OFACUEP emitirá un mensaje de aviso, si son incorrectos el sistema presentará un mensaje de error.

Req (04) OFACUEP permitirá eliminar los datos del Administrador Central, siempre y cuando estos datos existan y que otros datos no dependan de los que se van a eliminar.

Req(05) OFACUEP presentará un mensaje de aviso indicando que se eliminaron los datos, si no es así OFACUEP presentará un mensaje de aviso indicando que no se eliminaron.

Req (06) OFACUEP permitirá modificar los datos del Administrador Central, siempre y cuando estos datos existan y que otros datos no dependan de los que se van a modificar. Si otros datos dependen de los que se van a modificar, modificarlos también.

Req(07) OFACUEP presentará un mensaje de aviso indicando que se modificaron los datos, si no es así OFACUEP presentará un mensaje de aviso indicando que no se modificaron.

4.3.2 GESTIÓN UNIVERSIDADES

Req(08) El Administrador Universitario ingresará a la aplicación OFACUEP directamente desde el sitio web de la misma, para suscribir a la Universidad a la que representa.

Req(09) El administrador universitario ingresará en OFACUEP los datos informativos de la Universidad tales como: Número de RUC, Número de registro oficial otorgado por el CONESUP, Nombre de la Universidad, Siglas del Nombre de la Universidad, Nombre del Rector o Director, dirección del campus principal, teléfonos, e-mail,

página web, nombre del administrador universitario, e-mail del administrador universitario, estado(activo o inactivo).

Req(10) Una vez cumplido el Req(09), el administrador universitario registrará a su Universidad en OFACUEP enviando los datos en un formulario de suscripción.

Req(11) El administrador central verificará en forma manual, que la información enviada sea correcta.

Req(12) Si la información enviada es correcta, el Administrador activa a la Universidad asignándole un login y un password que serán únicos para cada Universidad, los mismos que envía por e-mail al Administrador Universitario.

Req(13) Si la información enviada no es correcta el administrador central le notificará al administrador universitario vía e-mail indicándole que se vuelva a registrar.

Req(14) Una vez cumplido el Req(12), el administrador Universitario debe acceder al sitio web de OFACUEP, ingresar el login y password

Req(15) Si el login y password son correctos el administrador universitario está habilitado para ingresar la información de la oferta académica de la Universidad.

Req(16) Si el login y el password no son correctos OFACUEP presentará un mensaje de error indicando que estos datos son incorrectos y se los debe ingresar nuevamente.

Req(17) Si cumple con el Req(15) El administrador universitario podrá ingresar los datos de las especialidades o carreras de acuerdo a título, modalidad, ciudad, nivel, duración, dirección, infraestructura, siendo primordial el ingreso de todos los datos

para que se puedan almacenar correctamente en el sistema, si no es así, OFACUEP presentará un mensaje de error, notificando que no se almacenaron los datos correctamente.

Req (18) El administrador Universitario podrá eliminar los datos de la Oferta académica siempre y cuando estos existan y que otros datos no dependan de los que se van a eliminar, pero no podrá eliminar los datos de la Universidad sino simplemente cambiar de estado activo a inactivo lo que indicará que esa universidad ya no forma parte de OFACUEP.

Req (19) OFACUEP presentará un mensaje de aviso indicando que se eliminaron los datos, si no es así OFACUEP presentará un mensaje de aviso indicando que no se eliminaron los datos.

Req (20) El administrador Universitario podrá modificar los datos de la Universidad y/o la Oferta académica siempre y cuando estos existan.

Req (21) Si otros datos dependen de los que se van a modificar, modificar también estos datos

Req (22) OFACUEP presentará un mensaje de aviso indicando que se modificaron los datos, si no es así OFACUEP presentará un mensaje de aviso indicando que no se eliminaron los datos.

4.3.3 BÚSQUEDA DE ESPECIALIDADES

Req (23) El usuario accederá a la Búsqueda de las carreras o especialidades desde el sitio web de OFACUEP.

Req (24) El sistema permitirá que el usuario realice búsquedas por:

- Nombre de la Universidad
- Nombre de la Especialidad O Carrera
- Título que se Oferta
- Modalidad (Presencial, Semipresencial o a Distancia)
- Ciudad
- Nivel (Pregrado o Postgrado)
- Dirección
- Login y password

Req (25) El resultado de la búsqueda se visualizará en el sitio de OFACUEP.

Req (26) Si el usuario realiza una búsqueda en la cual OFACUEP no arroja resultados, el sistema presentará un mensaje de aviso indicando que no hay resultados en la búsqueda realizada.

4.3.4 INTERFAZ UNIVERSIDAD

Req(27) El sistema Universidad envía a OFACUEP el criterio de búsqueda, la búsqueda es realizada por la Interfaz Universidad.

Req(28) La interfaz Universidad realiza la búsqueda y entrega la respuesta.

Req(29) Si el sistema Universidad envía un criterio de búsqueda con el cual OFACUEP no presenta resultados, el sistema presentará un mensaje de aviso indicando que no hay resultados en la búsqueda realizada.

4.4 REQUISITOS DE INTERFACES EXTERNAS

4.4.1 INTERFACES DE USUARIO

OFACUEP tendrá una interfaz de usuario orientada a ventanas y el manejo del sistema se realizará a través del teclado y el ratón.

4.4.2 INTERFACES DE SOFTWARE

OFACUEP requiere de una interfaz basada en XML para comunicarse con otros sistemas

4.4.3 INTERFACES DE COMUNICACIÓN

OFACUEP está orientado para funcionar en línea sobre Internet

4.5 REQUISITOS DE DESARROLLO

El ciclo de vida elegido para desarrollar el producto será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

4.6 REQUISITOS TECNOLÓGICOS DE HARDWARE Y SOFTWARE

	Cliente	Servidor
Sistema Operativo	Windows Me, Windows XP, Linux	Linux o Windows XP en adelante
Memoria (MB)	64 o superior	128 o superior
Procesador	Pentium III o superior	Pentium III o superior

Velocidad	800 Mhz o superior	800 Mhz o superior
Conexión a Internet	Dial Up como mínimo	128 Kbps simétrico
Browser	IE5.5 o superior (para windows)ó Mozilla (para Linux)	IE 5.5 o superior (para windows)ó Mozilla (para Linux)
Programas instalados	_____	PHP 5.0 Apache1.3.33
Dispositivos adicionales	_____	CD ROM

Tabla 4. 4

4.7 ATRIBUTOS

4.7.1 SEGURIDAD

El sistema permitirá el manejo de los datos de los clientes únicamente a las personas autorizadas, es decir a los administradores universitarios que poseen un login y un password, para que no se publiquen instituciones no autorizadas, el administrador central verificará manualmente si los datos de los clientes son correctos.

4.8 FASE DE ANÁLISIS

4.8.1 DIAGRAMAS DE CASOS DE USO

4.8.1.1 Ingresar Administrador Central

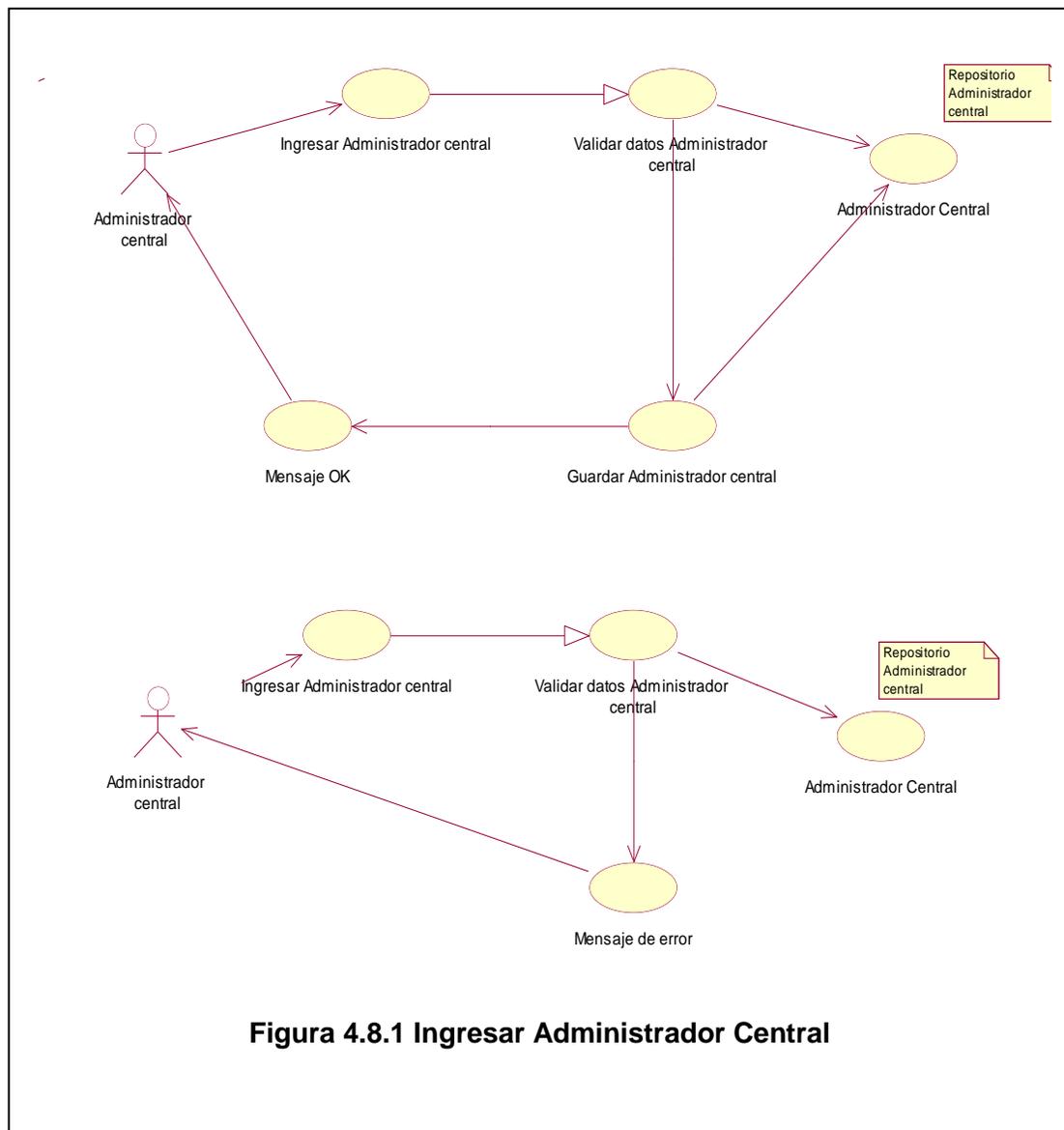


Figura 4.8.1 Ingresar Administrador Central

Actores: Administrador central (iniciador)

Propósito: Dar de alta a Administrador central

Precondiciones:

Resumen: Un administrador central ingresa a OFACUEP, introduce los datos de Administrador central y se da de alta en el Sistema

Al terminar la operación se da de alta a Administrador central

Tipo: Primario y esencial

Referencias cruzadas: Req01, Req02

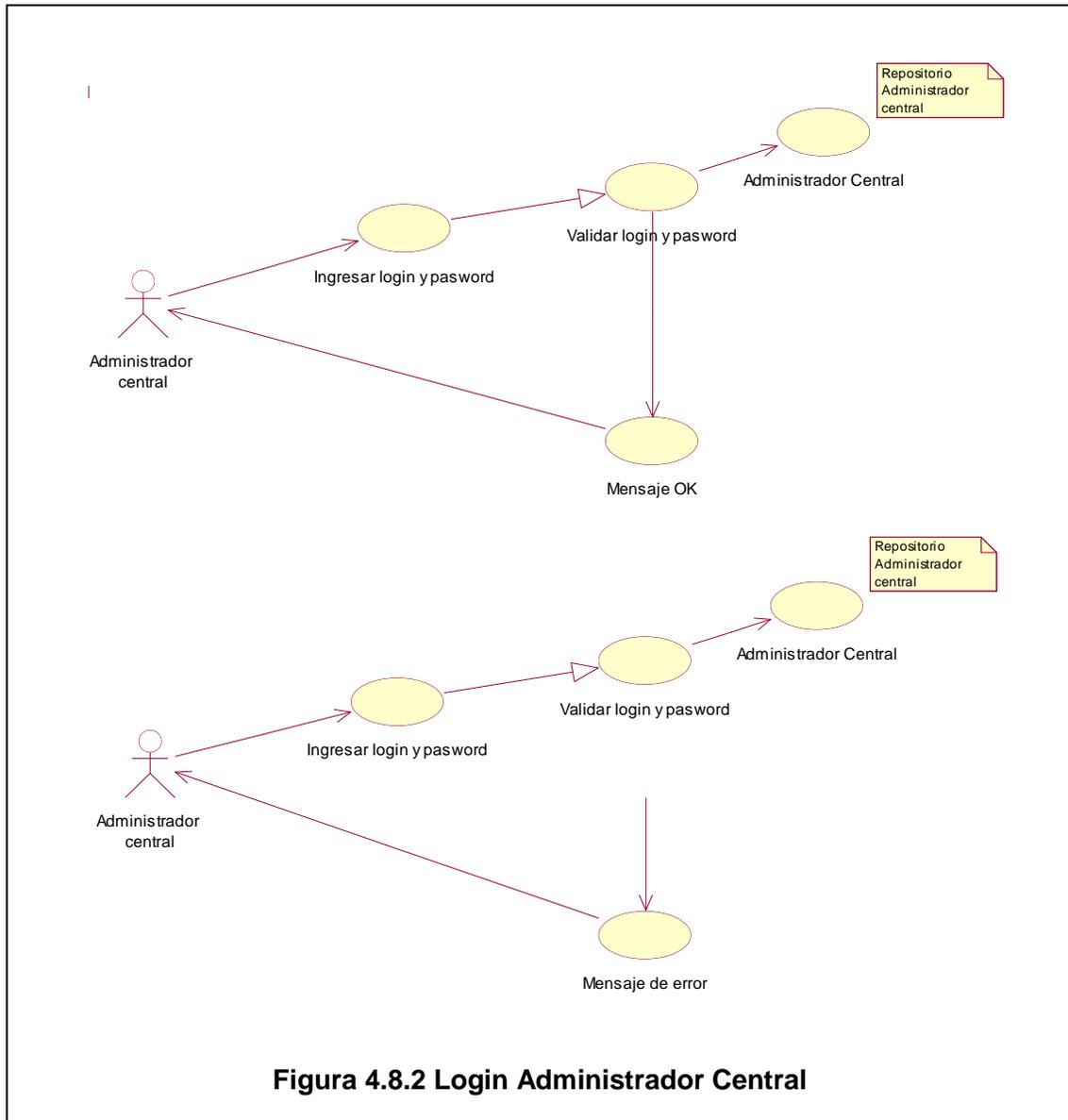
Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Ingresar Administrador central	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Validar datos Administrador central3. Guardar Administrador central4. Mensaje OK
---	--

Casos alternos:

Sección 3: Si los datos ingresados no son correctos, OFACUEP emite un Mensaje de error

4.8.1.2 Login Administrador central



Actores: Administrador central (iniciador)

Propósito: Autenticarse en el sistema

Precondiciones:

Resumen: Un administrador central se autentifica en OFACUEP, introduciendo su login y password.

Al terminar la operación se autentifica el Administrador central y puede administrar OFACUEP.

Tipo: Primario y esencial

Referencias cruzadas: Req03.

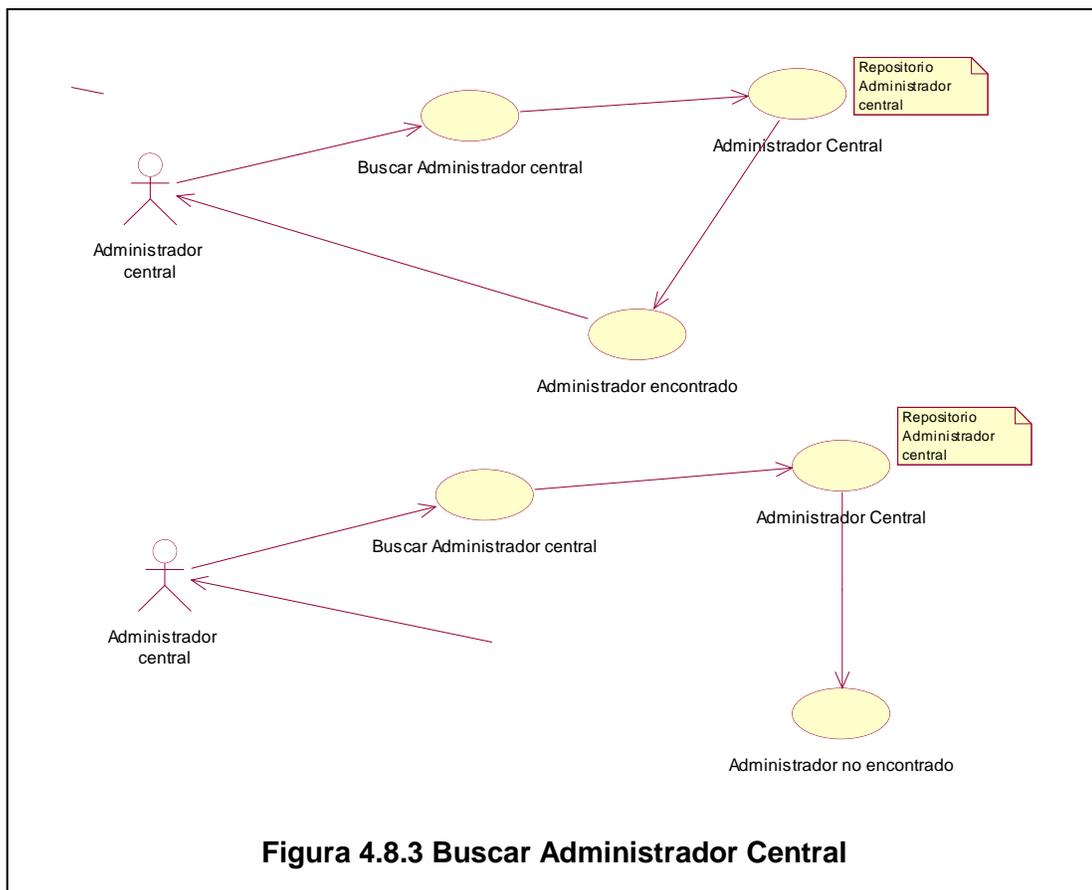
Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Ingresar login y password	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Validar login y password3. Mensaje OK
--	--

Casos alternos:

Sección 3: Si los datos ingresados no son correctos, OFACUEP emite un Mensaje de error

4.8.1.3 Buscar Administrador Central



Actores: Administrador central (iniciador)

Propósito: Buscar un Administrador central

Precondiciones: Petición eliminar o modificar

Resumen: Un administrador central busca en OFACUEP un registro de un administrador central.

Al terminar la operación se encuentra el registro buscado

Tipo: Primario y esencial

Referencias cruzadas: Req04, Req06

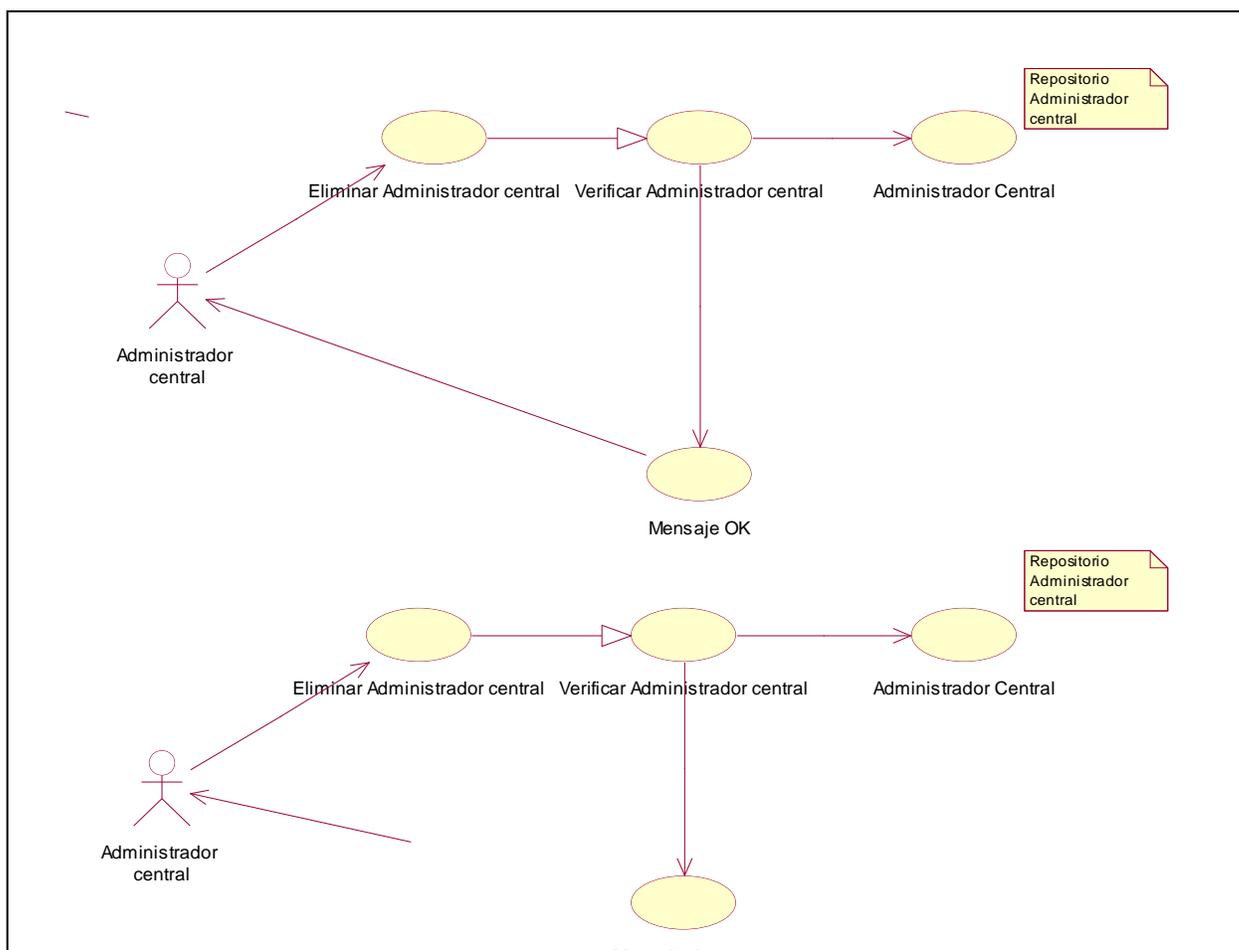
Curso normal de los eventos:

Acción de los actores: 1. Administrador central: Buscar Administrador central	Respuesta de OFACUEP: 2. Administrador encontrado
---	--

Casos alternos:

2. Si el administrador a buscar no es encontrado. Presentar un mensaje:
Administrador no encontrado

4.8.1.4 Eliminar Administrador central



Actores: Administrador central (iniciador)

Propósito: Eliminar un Administrador central

Precondiciones: Al buscar un Administrador central, éste debe existir

Resumen: El Administrador central elimina un registro de Administrador central.

Al terminar la operación se elimina el registro de Administrador central deseado

Tipo: Primario y esencial

Referencias cruzadas: Req04

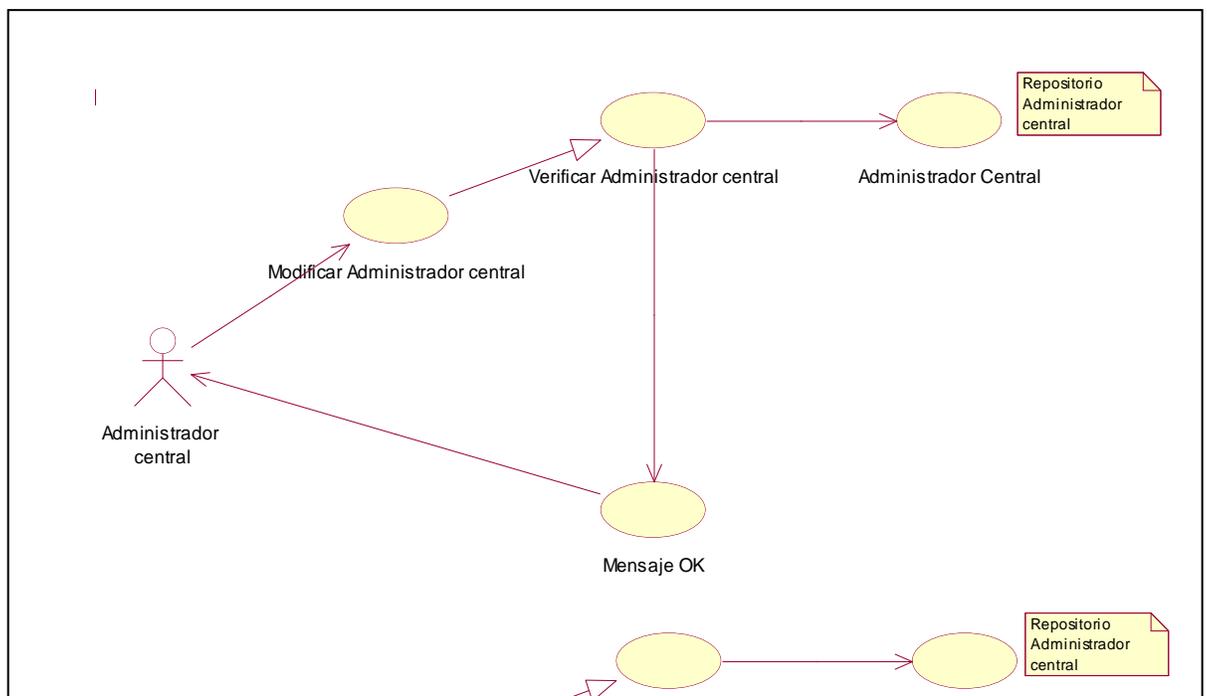
Curso normal de los eventos:

<p>Acción de los actores:</p> <p>1. Administrador central: Eliminar Administrador central</p>	<p>Respuesta de OFACUEP:</p> <p>2. Verificar Administrador central</p> <p>3. Mensaje OK</p>
---	---

Casos alternos:

3. Si al verificar administrador central se encuentra que no se puede eliminar este registro, OFACUEP presentará un Mensaje de Error

4.8.1.5 Modificar Administrador central



Actores: Administrador central (iniciador)

Propósito: Modificar un Administrador central

Precondiciones: Al buscar un Administrador central, éste debe existir

Resumen: El Administrador central modifica un registro de Administrador central.

Al terminar la operación se modifica el registro de Administrador central deseado

Tipo: Primario y esencial

Referencias cruzadas: Req06

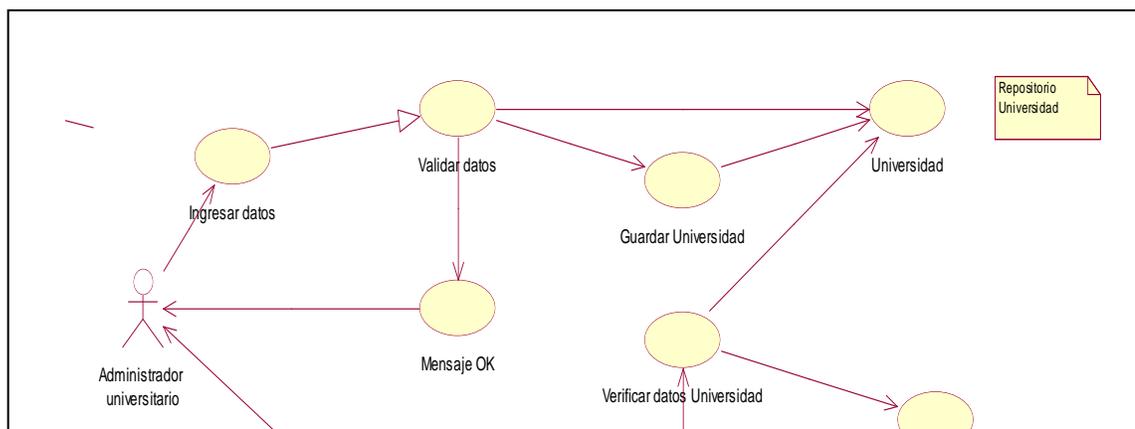
Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Administrador central: Modificar Administrador central	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Verificar Administrador central3. Mensaje OK
---	---

Casos alternos:

3. Si al verificar administrador central se encuentra que no se puede modificar este registro, OFACUEP presentará un Mensaje de Error.

4.8.1.6 Suscribir Universidad



Actores: Administrador universitario (iniciador), Administrador central.

Propósito: Suscribir Universidad.

Precondiciones:

Resumen: El Administrador universitario deberá ingresar los datos de su Universidad para que posteriormente sean verificados manualmente por el Administrador central quien y a través de un mail emitirá una respuesta de aceptación o rechazo.

Al terminar la operación la Universidad queda suscrita en OFACUEP

Tipo: Primario y esencial

Referencias cruzadas: Req09, Req10, Req11, Req12, Req13

Curso normal de los eventos:

Acción de los actores: 1. Administrador universitario: Ingresar datos 5. Verificar datos Universidad 6. Aceptar/Rechazar 7. Enviar respuesta	Respuesta de OFACUEP: 2. Validar datos 3. Mensaje OK 4. Guardar Universidad
--	--

Casos alternos:

3. Si al validar los datos no son correctos, OFACUEP presentará un Mensaje de Error.

4.8.1.7 Login Universidad

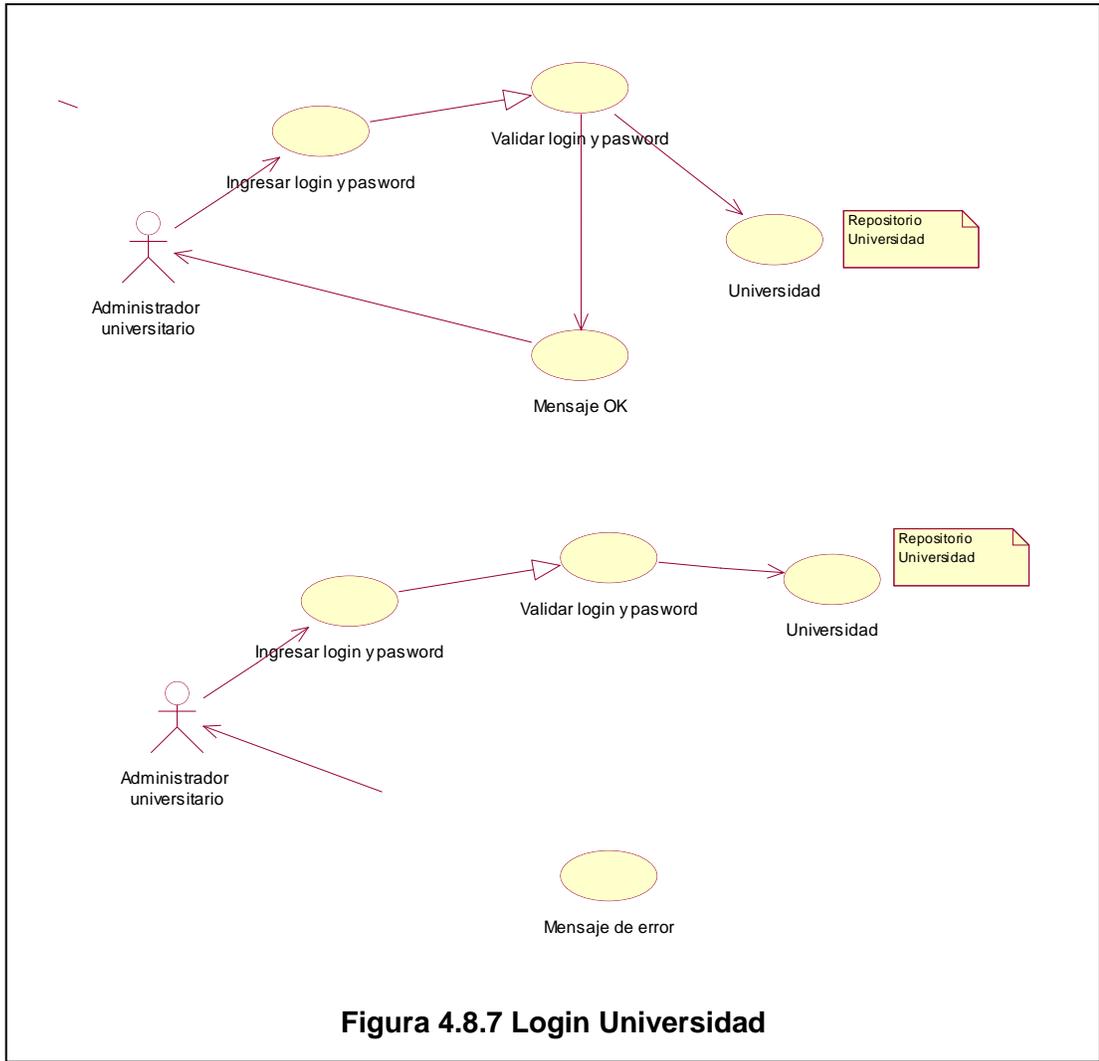


Figura 4.8.7 Login Universidad

Actores: Administrador Universitario

Propósito: Autenticar a la Universidad en OFACUEP.

Precondiciones:

Resumen: Un administrador Universitario autentifica a su Universidad en OFACUEP, introduciendo el login y password asignados

Al terminar la operación se autentifica a la Universidad, para proceder a introducir la oferta académica.

Tipo: Primario y esencial

Referencias cruzadas: Req14, Req15, Req16.

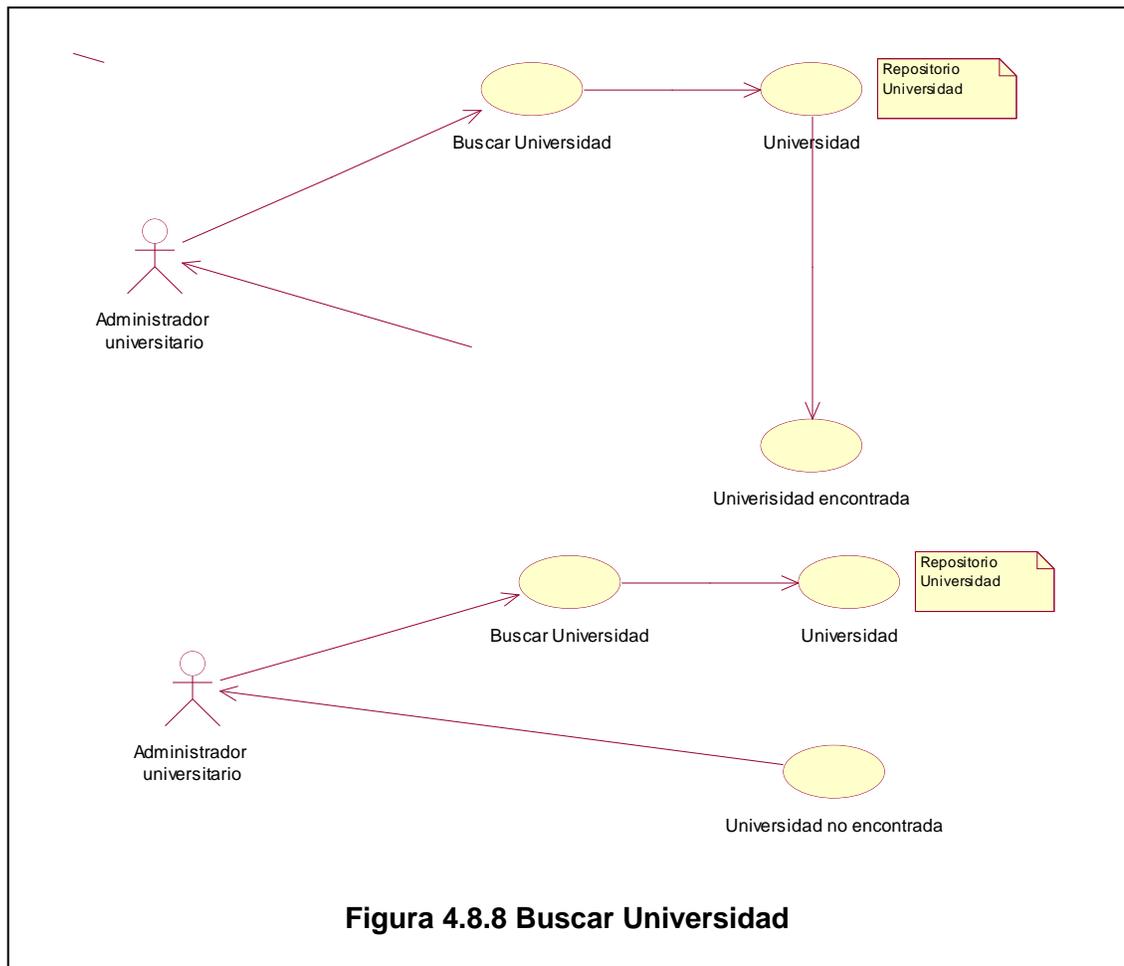
Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Ingresar login y password	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Validar login y password3. Mensaje OK
--	--

Casos alternos:

Sección 3: Si los datos ingresados no son correctos, OFACUEP emite un Mensaje de error

4.8.1.8 Buscar Universidad



Actores: Administrador universitario (iniciador)

Propósito: Buscar una Universidad

Precondiciones: Petición modificar

Resumen: Un administrador universitario busca en OFACUEP un registro de una Universidad.

Al terminar la operación se encuentra el registro buscado

Tipo: Primario y esencial

Referencias cruzadas: Req20

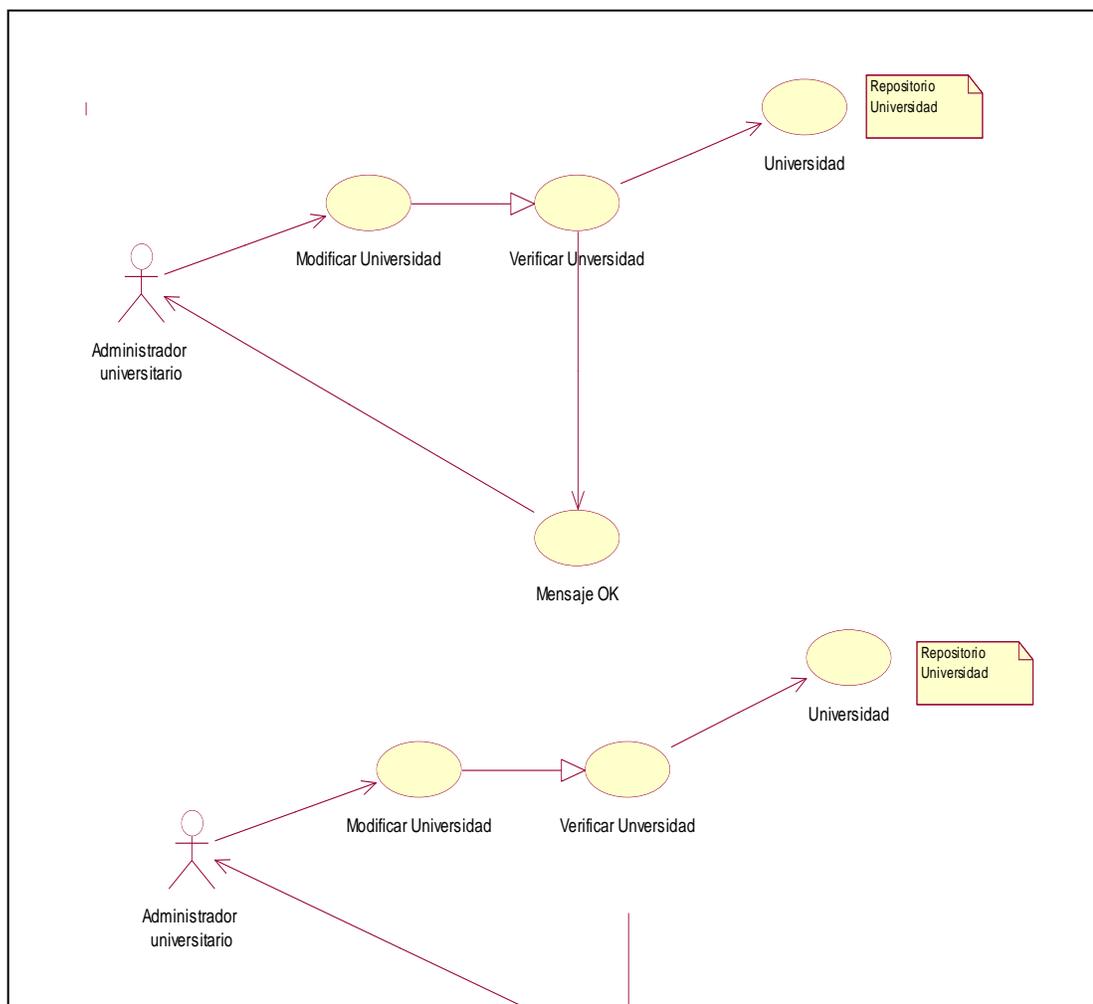
Curso normal de los eventos:

Acción de los actores: 1. Buscar Universidad	Respuesta de OFACUEP: 2. Universidad encontrada
---	--

Casos alternos:

2. Si la Universidad a buscar no es encontrada. Presentar un mensaje de error.

4.8.1.9 Modificar Universidad



Actores: Administrador universitario (iniciador)

Propósito: Modificar una Universidad

Precondiciones: Petición modificar

Resumen: Un administrador universitario busca en OFACUEP un registro de una Universidad.

Al terminar la operación se encuentra el registro buscado

Tipo: Primario y esencial

Referencias cruzadas: Req20, Req21, Req22.

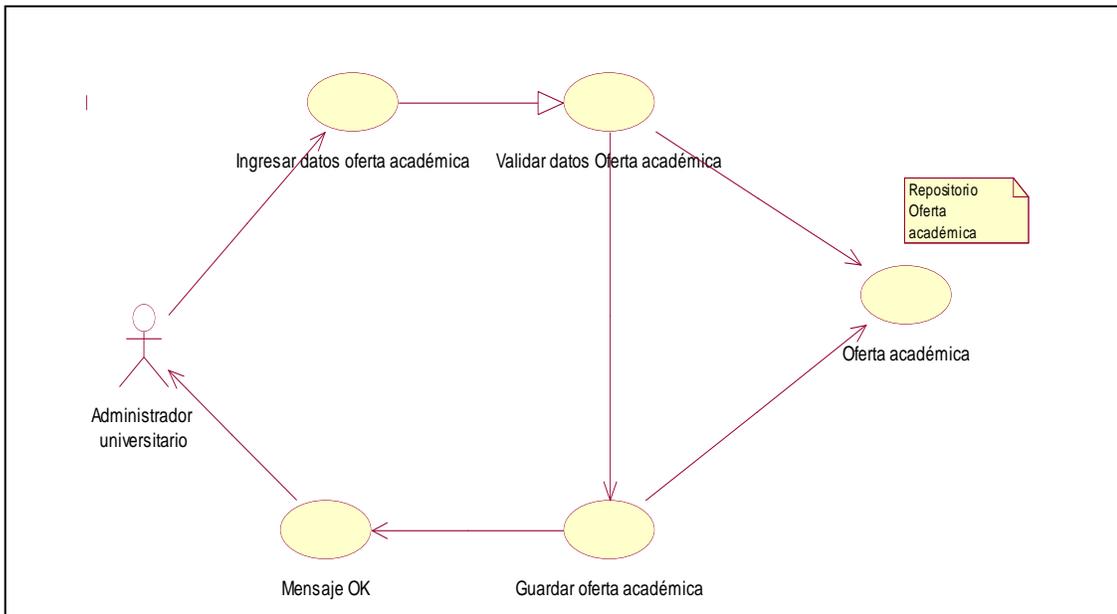
Curso normal de los eventos:

<p>Acción de los actores:</p> <p>1. Modificar Universidad</p>	<p>Respuesta de OFACUEP:</p> <p>2. Verificar Universidad</p> <p>3. Mensaje OK</p>
---	---

Casos alternos:

2. Si la Universidad a buscar no es encontrada. Presentar un mensaje de error.

4.8.1.10 Ingresar Oferta Académica



Actores: Administrador Universitario (iniciador)

Propósito: Dar de alta a Oferta Académica

Precondiciones: Login Universidad.

Resumen: Un administrador universitario ingresa a OFACUEP, introduce los datos de Oferta académica y la da de alta en el sistema

Al terminar la operación se da de alta a Oferta académica

Tipo: Primario y esencial

Referencias cruzadas: Req17.

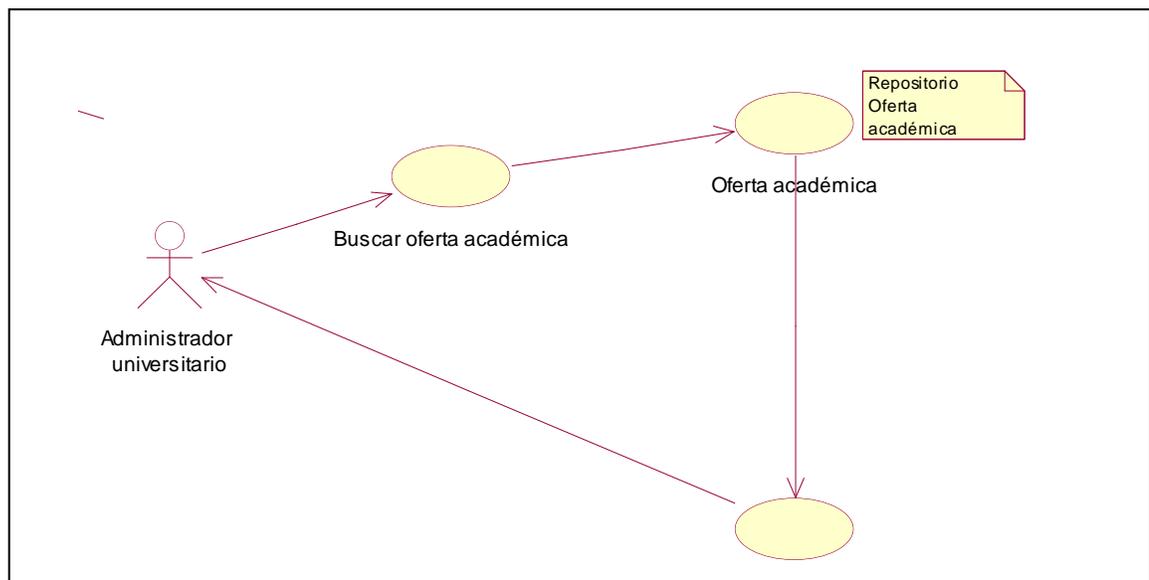
Curso normal de los eventos:

Acción de los actores: 1. Ingresar datos oferta académica	Respuesta de OFACUEP: 2. Validar Oferta académica 3. Guardar Oferta académica 4. Mensaje OK
--	--

Casos alternos:

Sección 3: Si no se guardan los datos de la Oferta académica, querrá decir que no cumplió con validar Oferta Académica, por lo que OFACUEP emitirá un Mensaje de error

4.8.1.11 Buscar Oferta Académica



Actores: Administrador universitario (iniciador)

Propósito: Buscar una Oferta Académica

Precondiciones: Petición eliminar o modificar Oferta Académica

Resumen: Un administrador universitario busca en OFACUEP un registro de Oferta académica.

Al terminar la operación se encuentra el registro buscado

Tipo: Primario y esencial

Referencias cruzadas: Req18, Req20

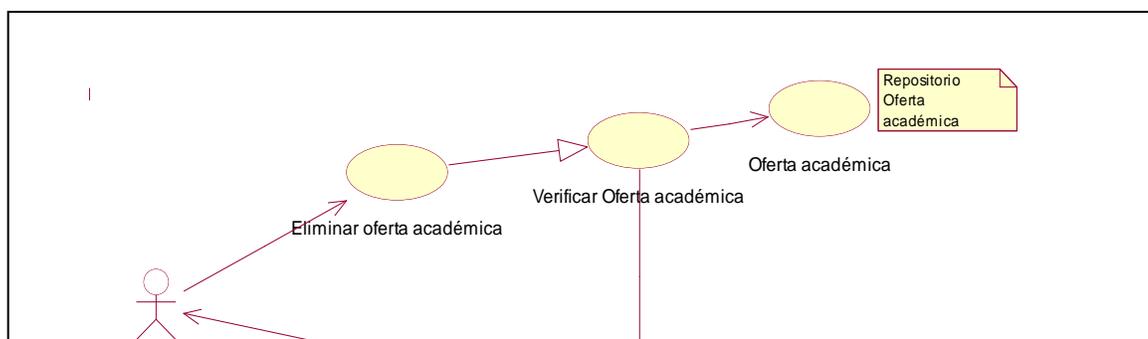
Curso normal de los eventos:

Acción de los actores: 1. Buscar Oferta académica	Respuesta de OFACUEP: 2. Oferta encontrada
--	---

Casos alternos:

2. Si el la oferta a buscar no es encontrada. Presentar un mensaje: Oferta no encontrada

4.8.1.12 Eliminar Oferta Académica



Actores: Administrador universitario (iniciador)

Propósito: Eliminar una Oferta académica

Precondiciones: Al buscar una Oferta académica, ésta debe existir

Resumen: El Administrador universitario elimina un registro de Administrador universitario.

Al terminar la operación se elimina el registro de Oferta académica deseada

Tipo: Primario y esencial

Referencias cruzadas: Req18, Req19.

Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Eliminar Oferta académica	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Validar Oferta académica3. Mensaje OK
--	--

Casos alternos:

3. Si al verificar que la Oferta Académica no puede eliminarse, OFACUEP presentará un Mensaje de Error

4.8.1.13 Modificar Oferta Académica

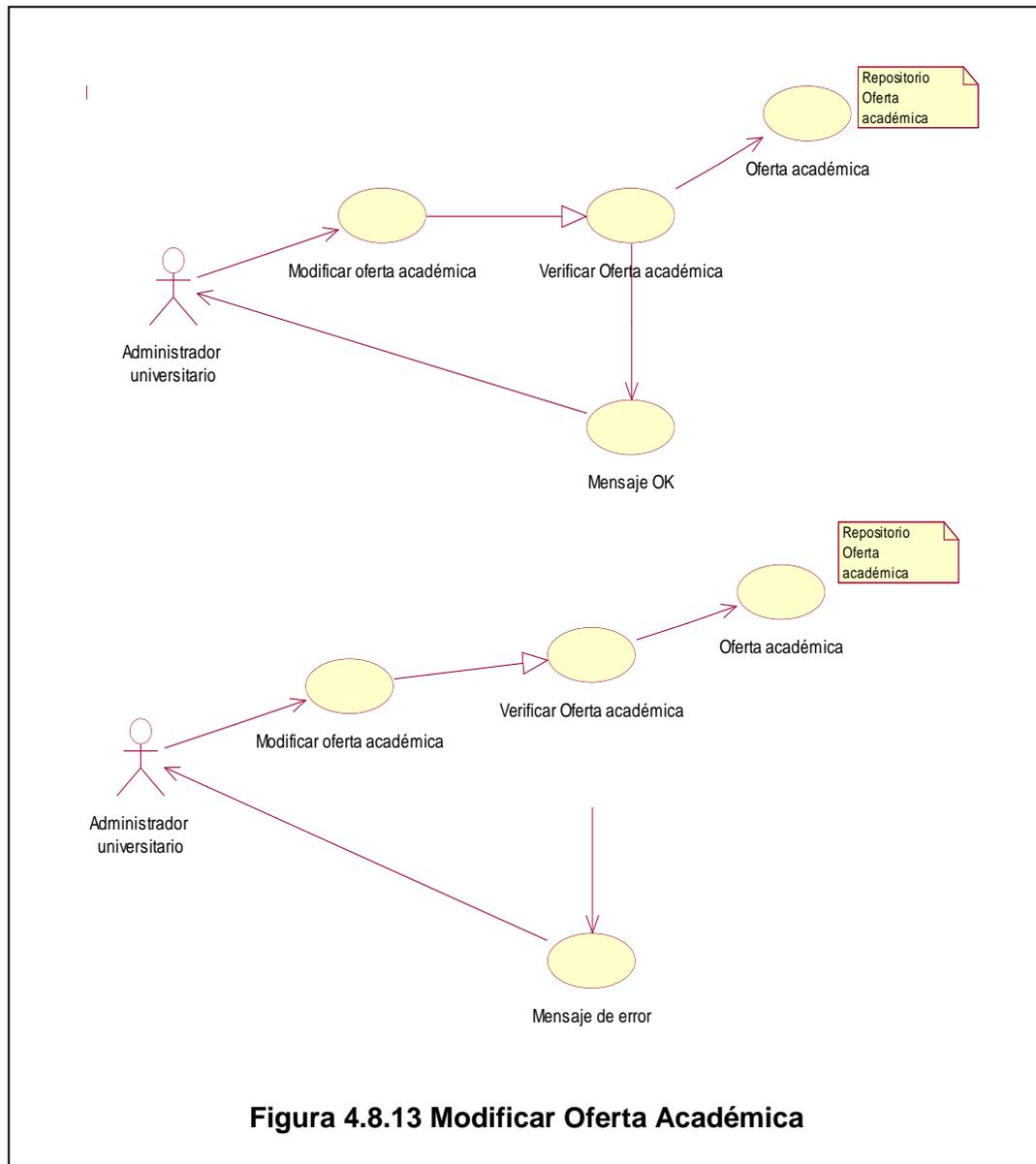


Figura 4.8.13 Modificar Oferta Académica

Actores: Administrador universitario (iniciador)

Propósito: Eliminar Oferta académica

Precondiciones: Al buscar una Oferta académica, ésta debe existir

Resumen: El Administrador universitario elimina un registro de Administrador universitario.

Al terminar la operación se elimina el registro de Oferta académica deseada

Tipo: Primario y esencial

Referencias cruzadas: Req18, Req19.

Curso normal de los eventos:

<p>Acción de los actores:</p> <ol style="list-style-type: none">1. Eliminar Oferta académica	<p>Respuesta de OFACUEP:</p> <ol style="list-style-type: none">2. Validar Oferta académica3. Mensaje OK
--	--

Casos alternos:

3. Si al verificar que la Oferta Académica no puede eliminarse, OFACUEP presentará un Mensaje de Error

4.8.1.14 Buscar Especialidades

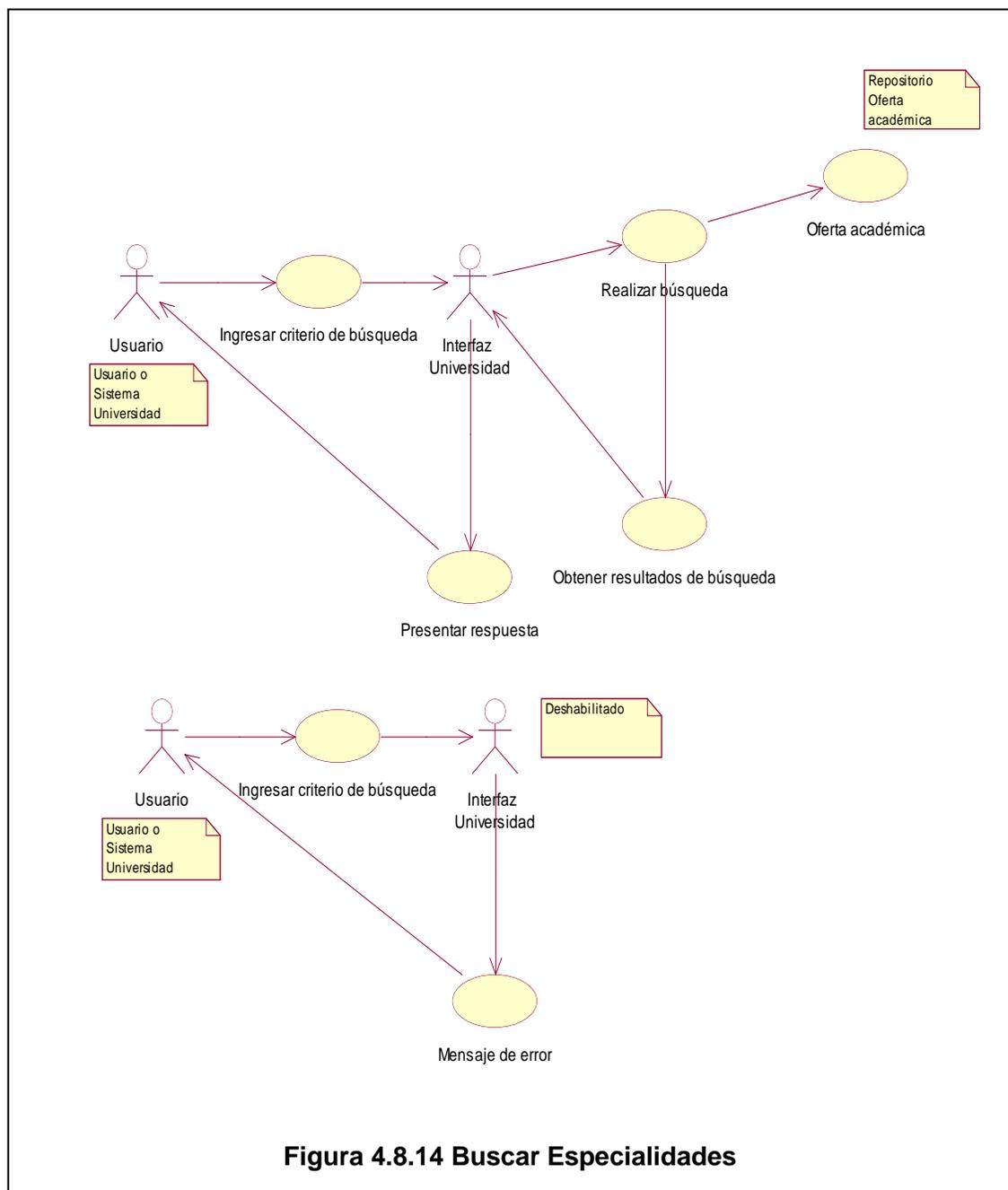


Figura 4.8.14 Buscar Especialidades

Actores: Administrador universitario (iniciador)

Propósito: Buscar Especialidades por el usuario

Precondiciones:

Resumen: El Administrador universitario elimina un registro de Administrador universitario

Al terminar la operación se elimina el registro de la Oferta académica deseada

Tipo: Primario y esencial

Referencias cruzadas: Req18, Req19.

Curso normal de los eventos:

Acción de los actores: 1. Eliminar Oferta académica	Respuesta de OFACUEP: 2. Validar Oferta académica 3. Mensaje OK
--	---

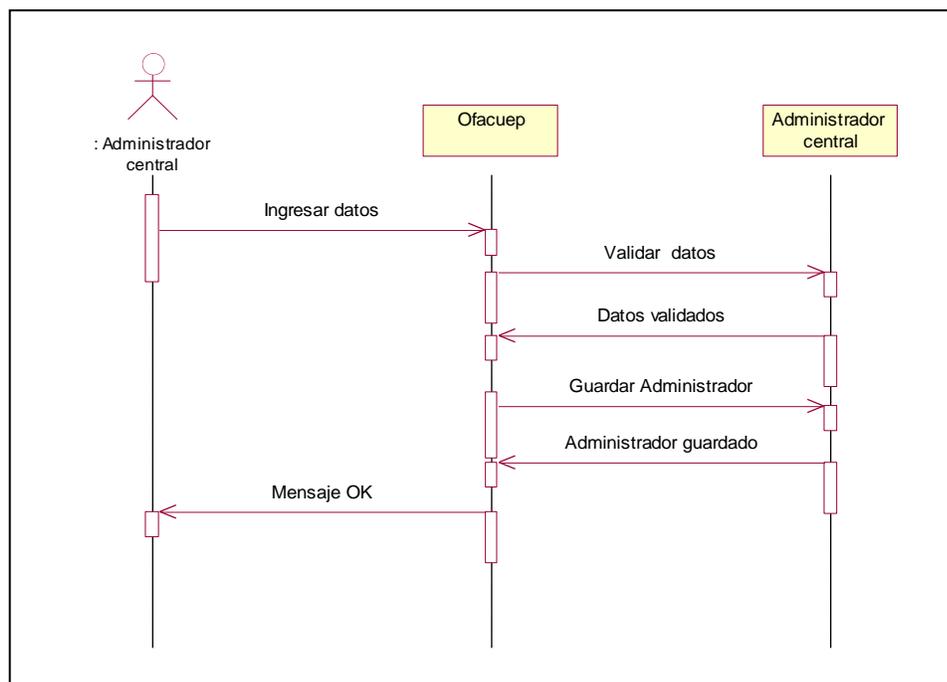
Casos alternos:

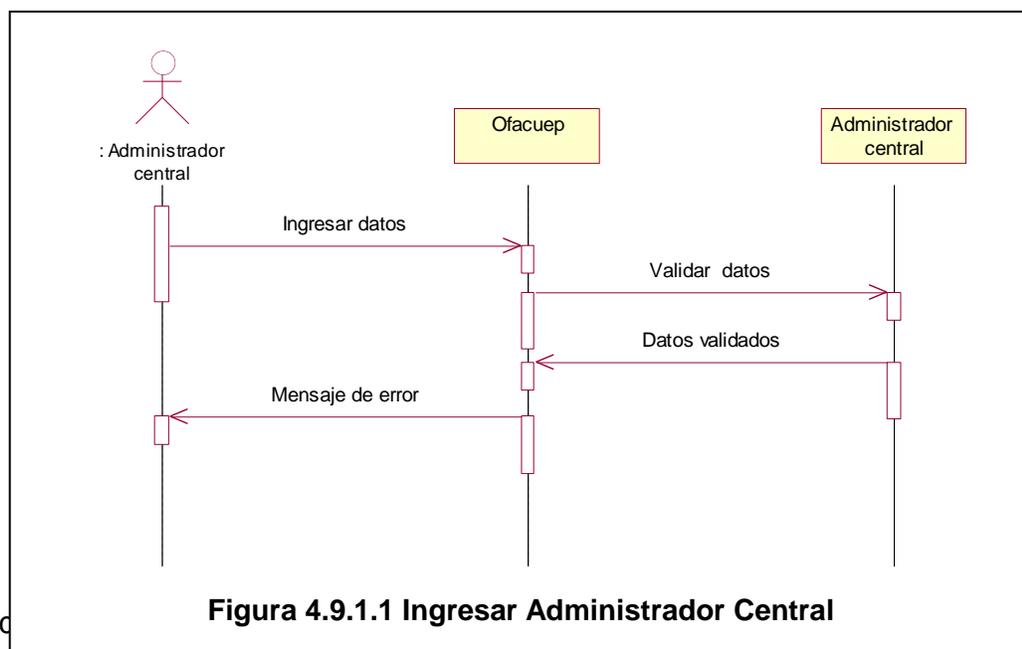
3. Si al verificar la Oferta Académica no puede eliminarse, OFACUEP presentará un Mensaje de Error.

4.9 FASE DE DISEÑO

4.9.1 DIAGRAMAS DE SECUENCIA

4.9.1.1 Ingresar Administrador Central





Validar d

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos, validar en base al campo cedulaadm si ya existe el registro a insertar, además comprobará que todos los campos se inserten.

Tipo: OFACUEP

Caso de uso: Ingresar administrador central

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Administrador

Nombre: Guardar administrador

Responsabilidades: Deberá guardar los datos del administrador central en el repositorio Administrador central

Tipo: OFACUEP

Caso de uso: Ingresar administrador central

Notas:

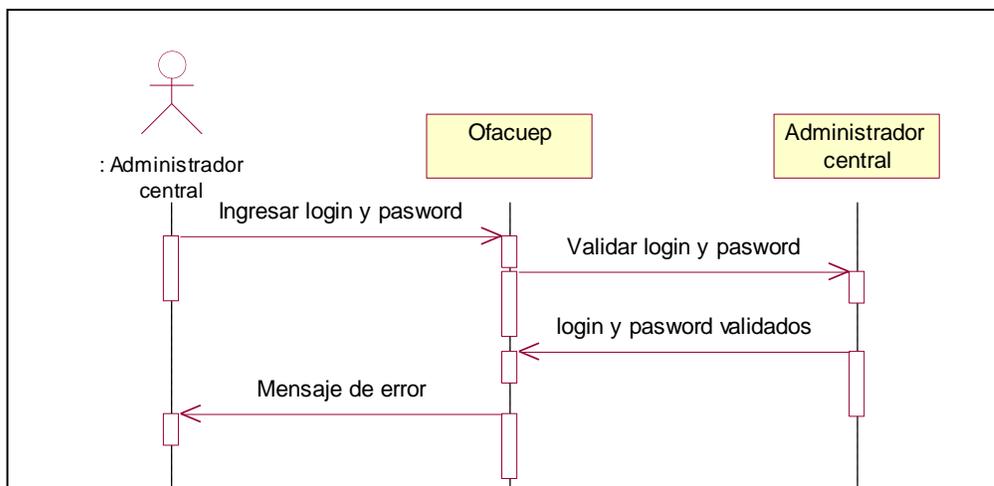
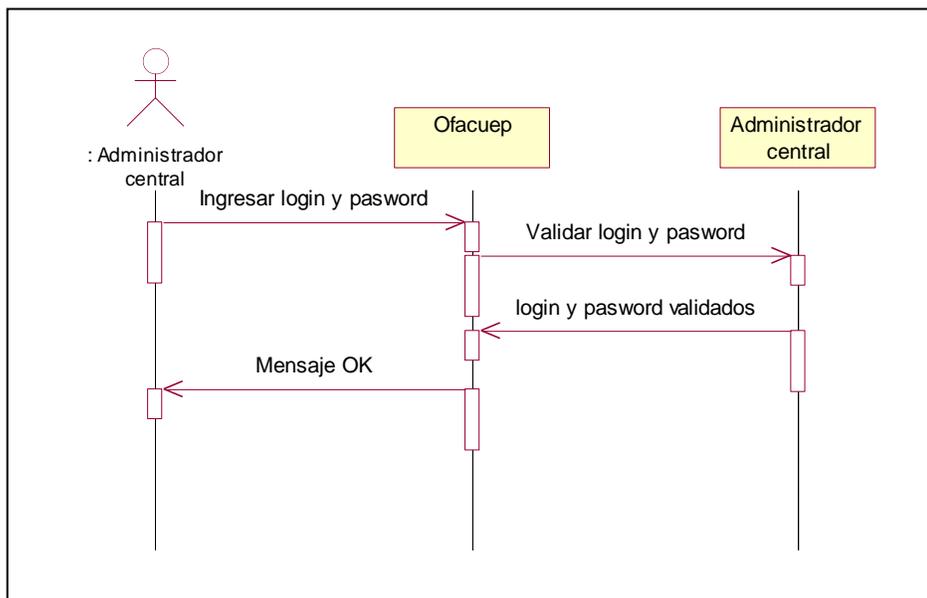
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Administrador guardado

4.9.1.2 Login Administrador Central



Validar login y password

Nombre: Validar login y password

Responsabilidades: Deberá comprobar que el login y password ingresados son auténticos.

Tipo: OFACUEP

Caso de uso: Login Administrador central

Notas:

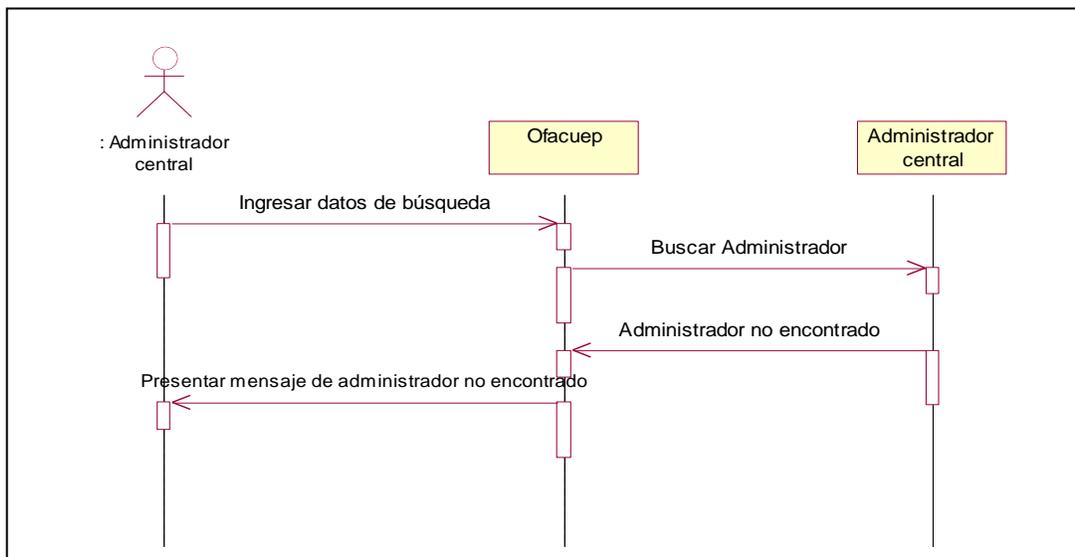
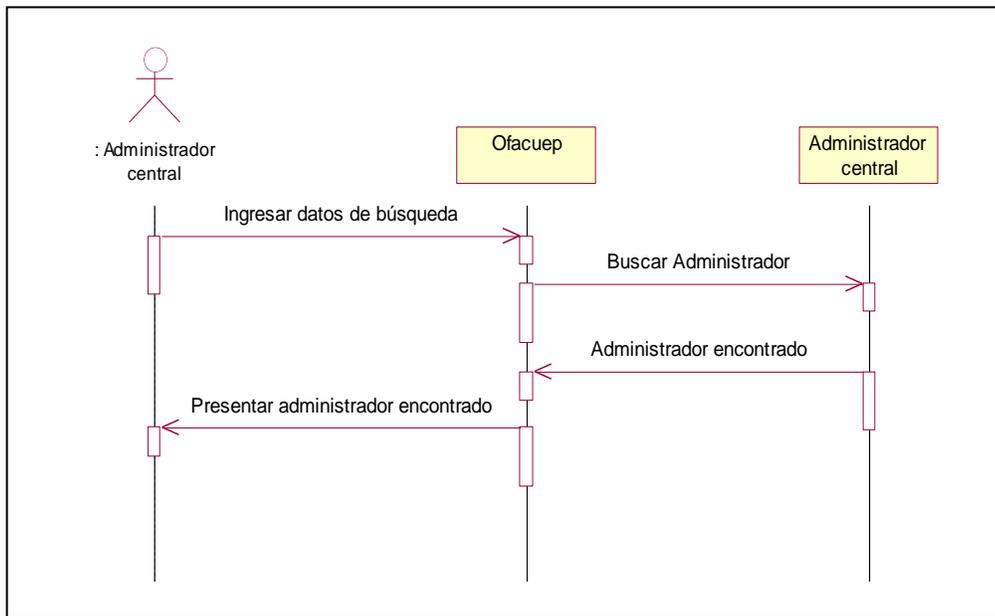
Excepciones:

Salidas:

Precondiciones: Ingresar login y password

Postcondiciones: login y password validados.

4.9.1.3 Buscar Administrador Central



Buscar Administrador

Nombre: Buscar administrador

Responsabilidades: Deberá buscar el registro del administrador central de acuerdo a los datos de búsqueda

Tipo: OFACUEP

Caso de uso: Buscar Administrador central

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos de búsqueda

Postcondiciones: Administrador encontrado o no encontrado

Presentar administrador encontrado

Nombre: Presentar administrador encontrado

Responsabilidades: Deberá presentar en pantalla el registro del administrador buscado

Tipo: OFACUEP

Caso de uso: Buscar Administrador central

Notas:

Excepciones:

Salidas:

Precondiciones: Buscar administrador

Postcondiciones:

Presentar msg administrador no encontrado

Nombre: Presentar msg administrador no encontrado

Responsabilidades: Deberá presentar en pantalla un mensaje indicando que no se encontró el registro a buscar

Tipo: OFACUEP

Caso de uso: Buscar Administrador central

Notas:

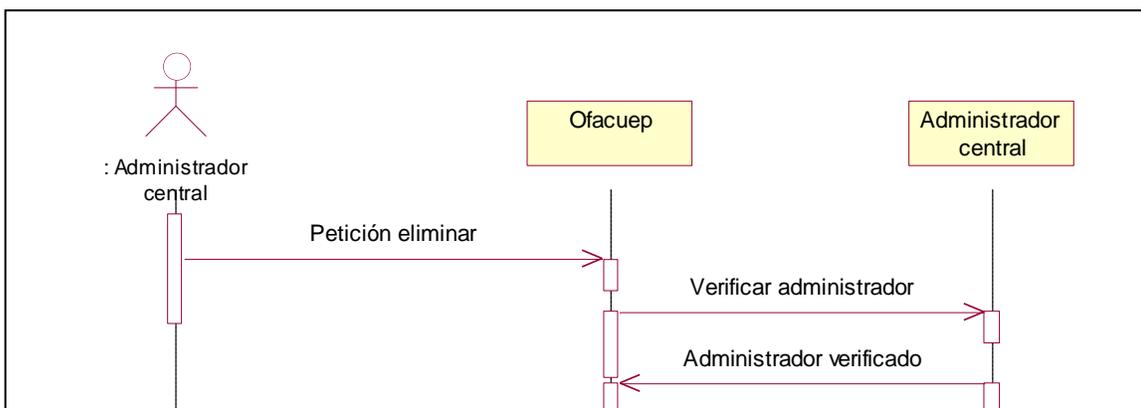
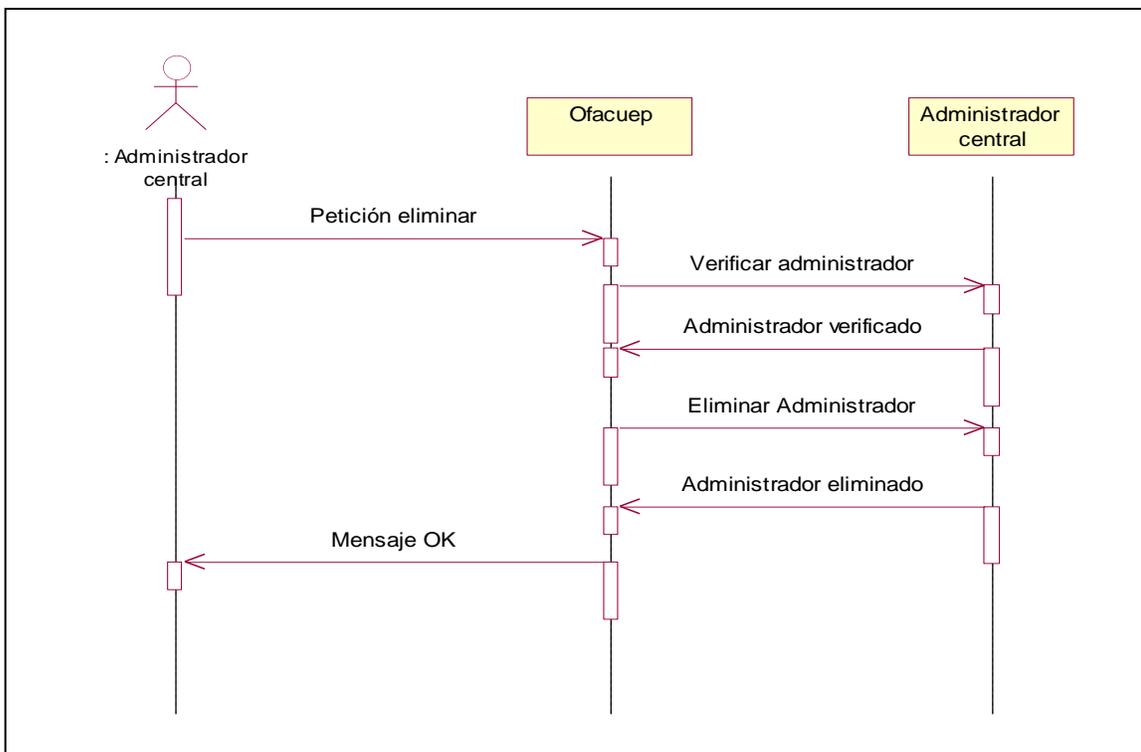
Excepciones:

Salidas:

Precondiciones: Buscar administrador

Postcondiciones:

4.9.1.4 Eliminar Administrador central



Verificar administrador

Nombre: Verificar administrador

Responsabilidades: Deberá verificar en base al cedulaadm, si los datos del administrador a eliminar tienen o no otros datos que dependan de ellos.

Tipo: OFACUEP

Caso de uso: Eliminar Administrador central

Notas:

Excepciones:

Salidas:

Precondiciones: Petición eliminar

Postcondiciones: Administrador verificado

Eliminar administrador

Nombre: Eliminar administrador

Responsabilidades: Deberá eliminar el administrador solicitado del repositorio Administrador central

Tipo: OFACUEP

Caso de uso: Eliminar Administrador central

Notas:

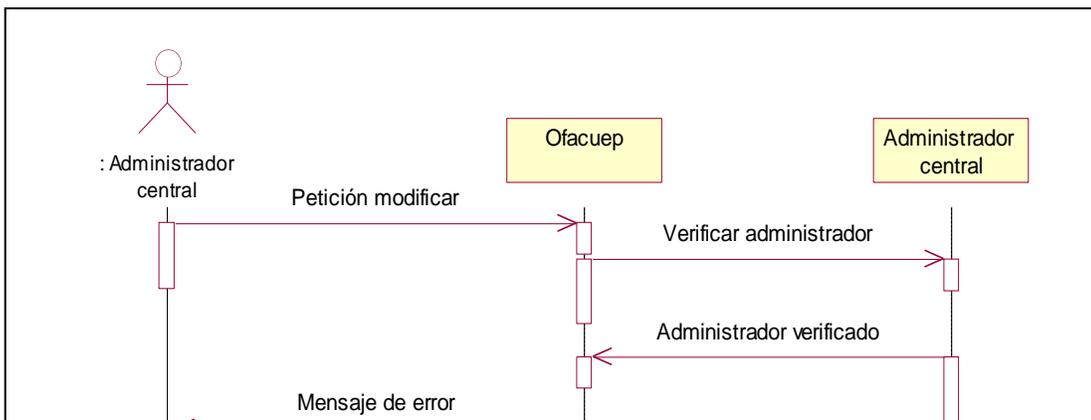
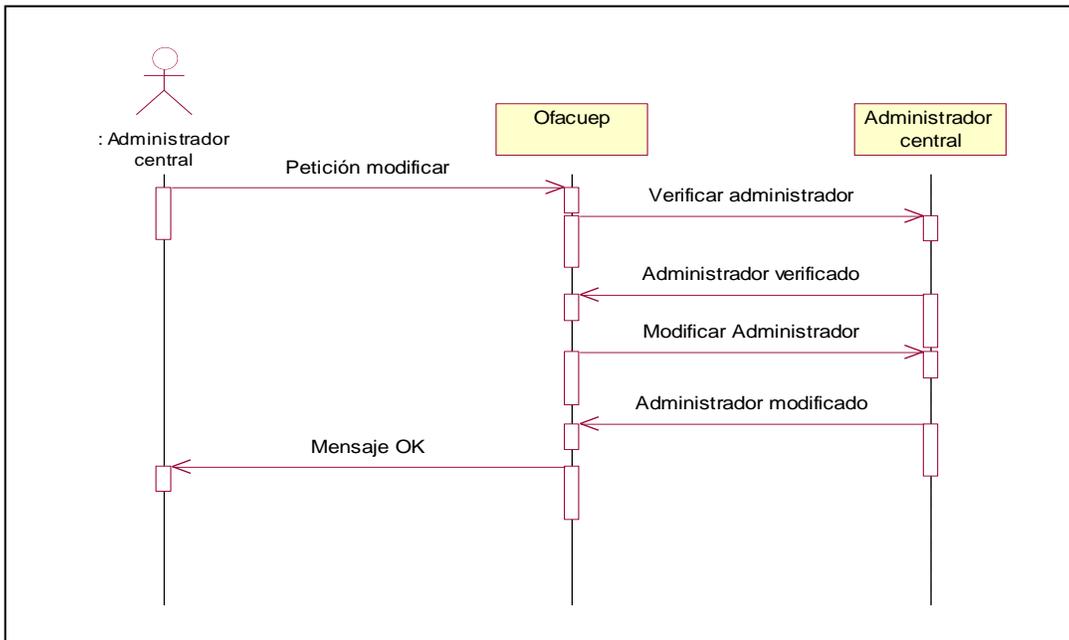
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Administrador eliminado

4.9.1.5 Modificar Administrador central



Verificar administrador

Nombre: Verificar administrador

Responsabilidades: Deberá verificar en base al campo cedulaadm, si los datos del administrador a modificar tienen o no otros datos que dependan de ellos

Tipo: OFACUEP

Caso de uso: Modificar Administrador central

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Administrador verificado

Modificar administrador

Nombre: Modificar administrador

Responsabilidades: Deberá modificar el administrador solicitado del repositorio Administrador central

Tipo: OFACUEP

Caso de uso: Modificar Administrador central

Notas:

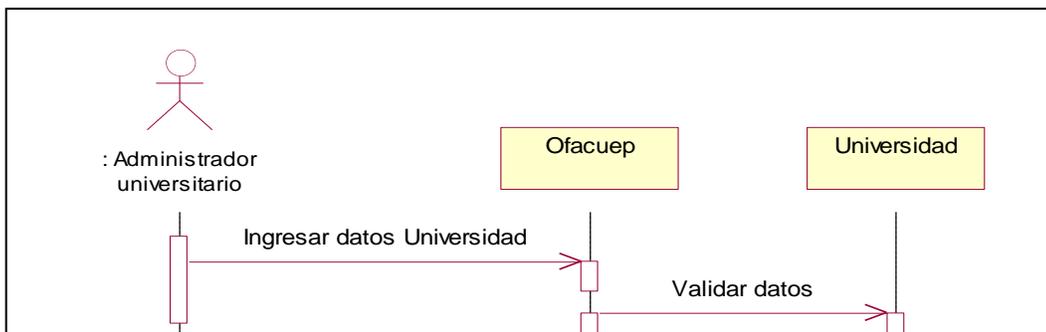
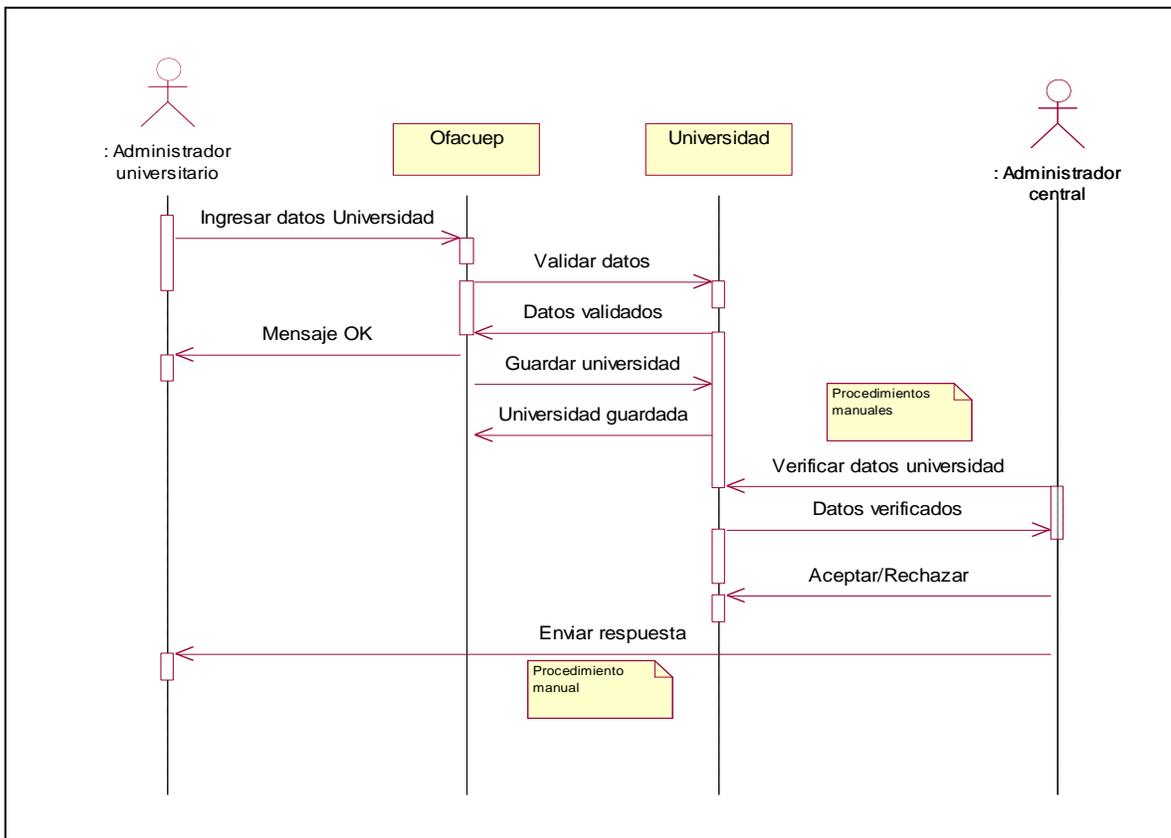
Excepciones:

Salidas:

Precondiciones: Verificar datos

Postcondiciones: Administrador modificado

4.9.1.6 Suscribir Universidad



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar el formato y tipo de los datos al momento de ingresarlos, validar en base al campo rucuni si ya existe el registro a insertar, además comprobará que todos los campos se inserten.

Tipo: OFACUEP

Caso de uso: Suscribirse Universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar Universidad

Nombre: Guardar universidad

Responsabilidades: Deberá guardar los datos de la Universidad en el repositorio Universidad.

Tipo: OFACUEP

Caso de uso: Suscribirse Universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Verificar datos Universidad

Postcondiciones: universidad guardada

4.9.1.7 Login Universidad

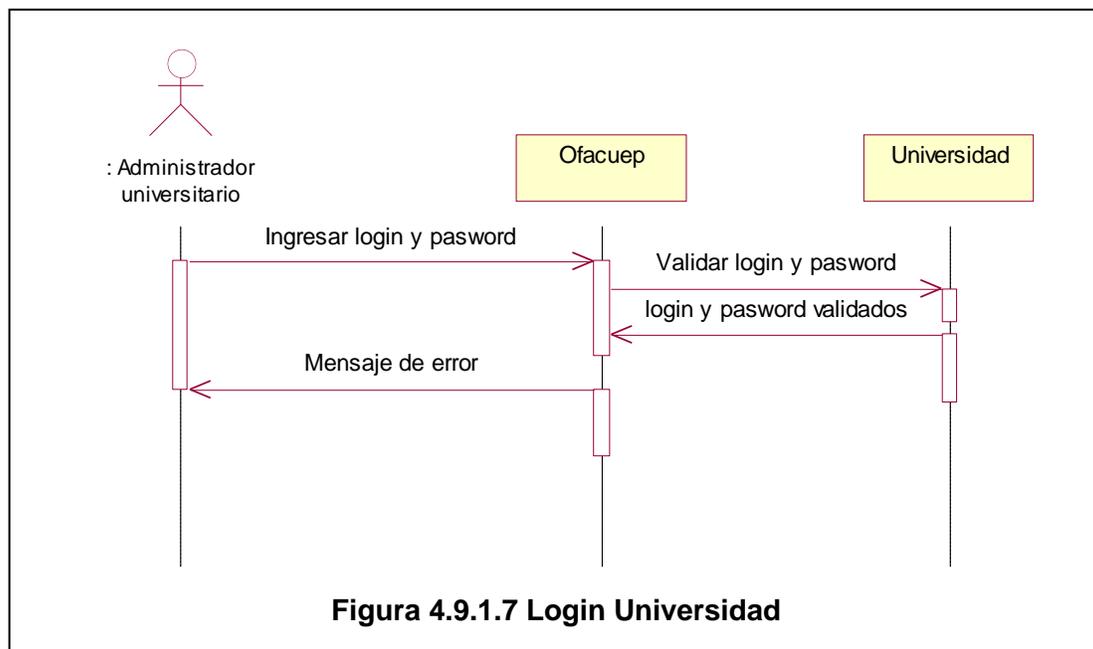
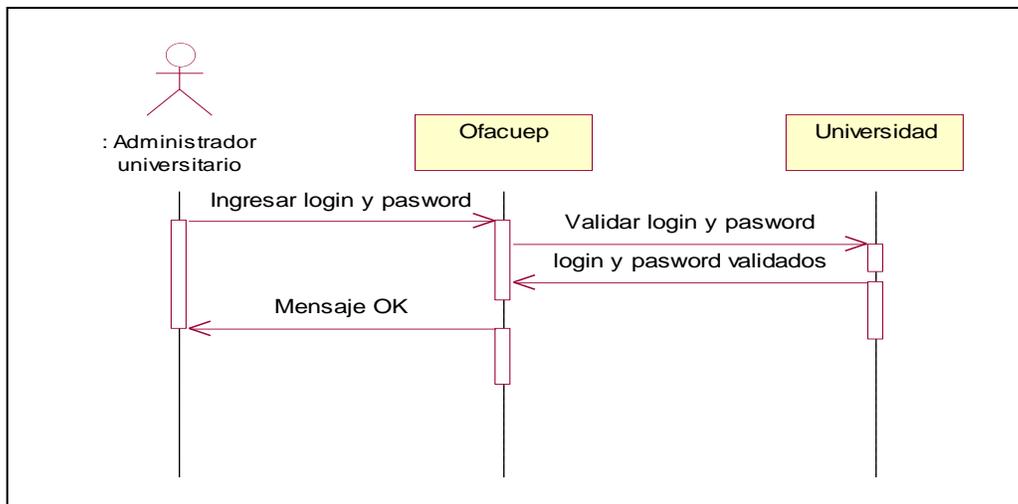


Figura 4.9.1.7 Login Universidad

Validar login y password

Nombre: Validar login y password

Responsabilidades: Deberá comprobar que el login y password ingresados son auténticos

Tipo: OFACUEP

Caso de uso: Login Universidad

Notas:

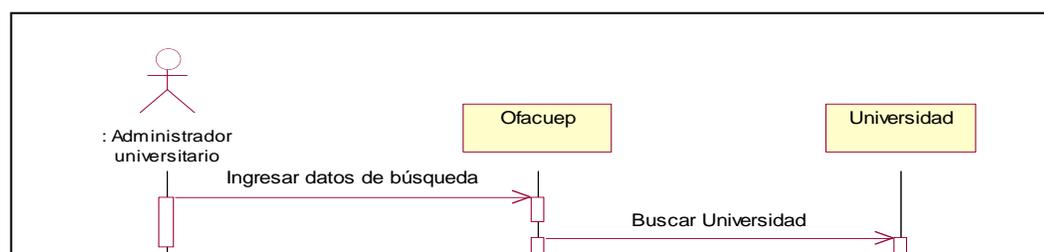
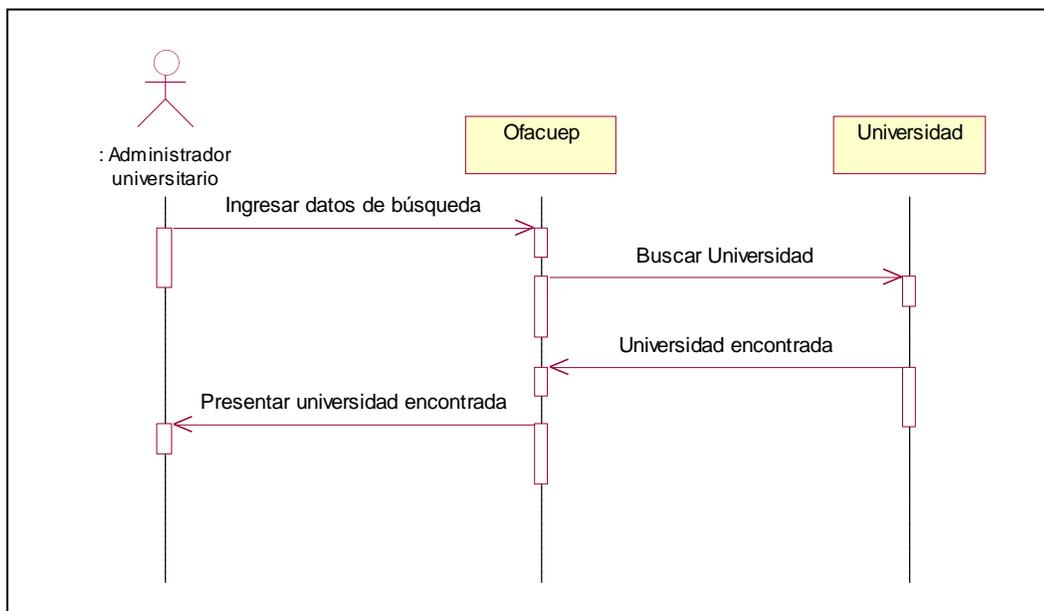
Excepciones:

Salidas:

Precondiciones: Ingresar login y password

Postcondiciones: login y password validados.

4.9.1.8 Buscar Universidad



Buscar Universidad

Nombre: Buscar Universidad

Responsabilidades: Deberá buscar el registro de la Universidad de acuerdo a los datos de búsqueda

Tipo: OFACUEP

Caso de uso: Buscar Universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos de búsqueda

Postcondiciones: Universidad encontrada o no encontrada

Presentar universidad encontrada

Nombre: Presentar universidad encontrada

Responsabilidades: Deberá presentar en pantalla universidad buscada

Tipo: OFACUEP

Caso de uso: Buscar universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Buscar universidad

Postcondiciones:

Presentar msg universidad no encontrada

Nombre: Presentar msg universidad no encontrada

Responsabilidades: Deberá presentar en pantalla un mensaje indicando que no se encontró la universidad a buscar

Tipo: OFACUEP

Caso de uso: Buscar universidad

Notas:

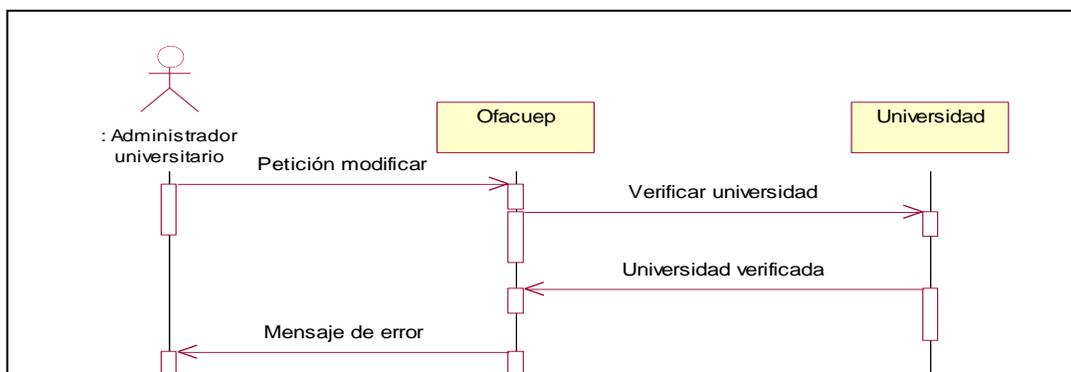
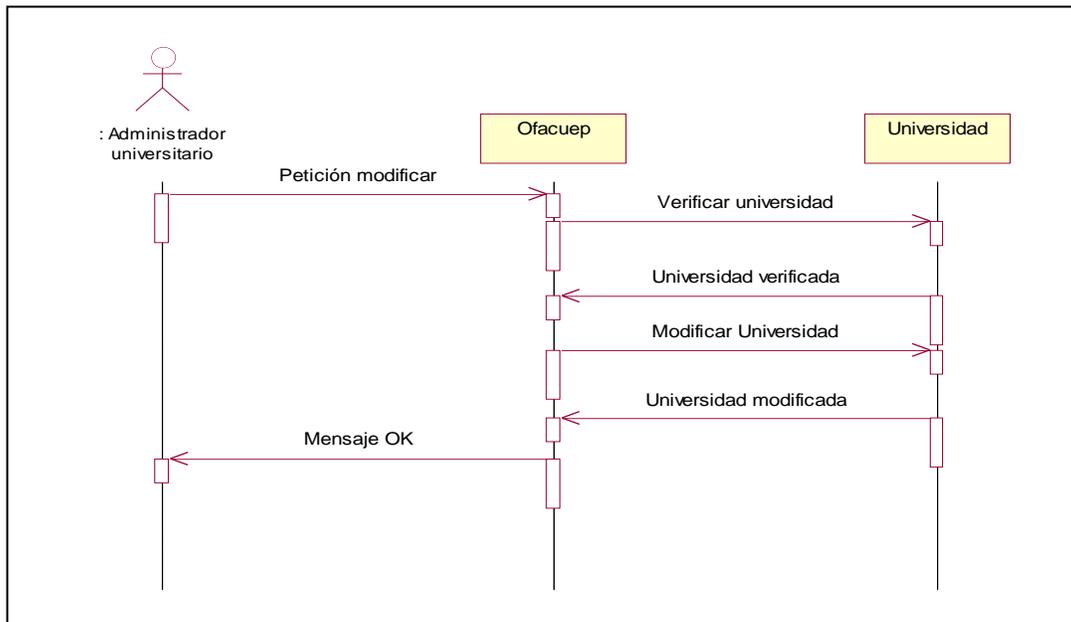
Excepciones:

Salidas:

Precondiciones: Buscar universidad

Postcondiciones:

4.9.1.9 Modificar Universidad



Verificar universidad

Nombre: Verificar universidad

Responsabilidades: Deberá verificar en base al campo rucuni, si los datos a modificar tienen o no otros datos que dependan de ellos.

Tipo: OFACUEP

Caso de uso: Modificar universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Universidad verificada

Modificar universidad

Nombre: Modificar universidad

Responsabilidades: Deberá modificar la universidad solicitado del repositorio Universidad

Tipo: OFACUEP

Caso de uso: Modificar universidad

Notas:

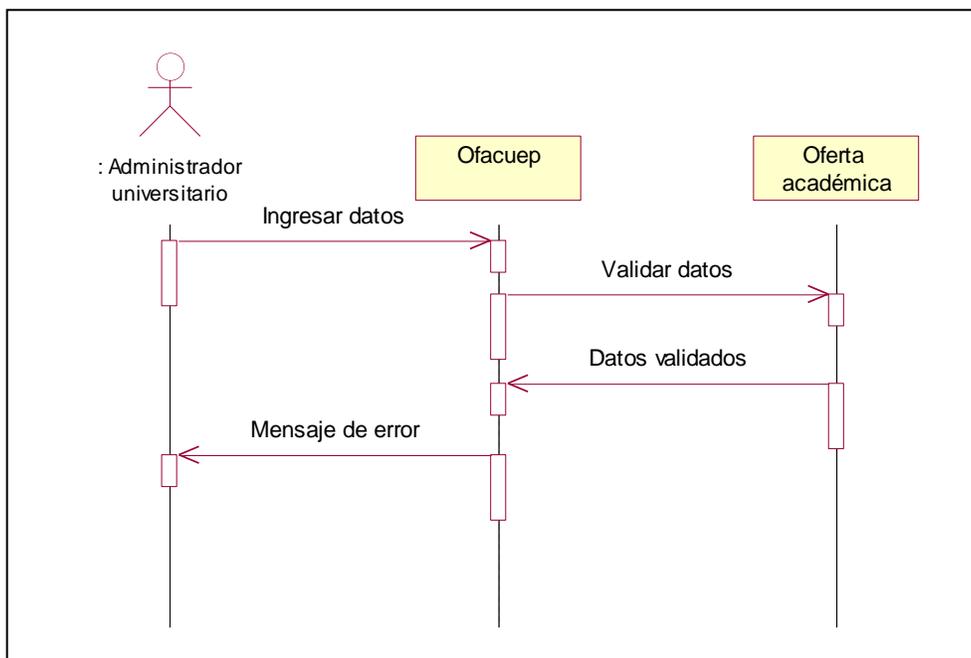
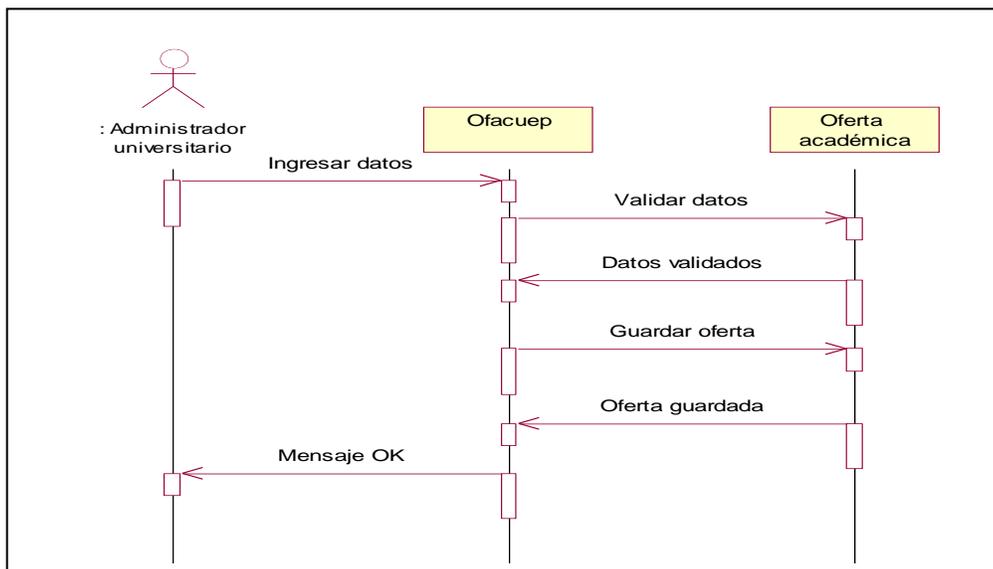
Excepciones:

Salidas:

Precondiciones: Verificar universidad

Postcondiciones: Universidad modificada

4.9.1.10 Ingresar Oferta Académica



Validar datos

Nombre: Validar datos

Responsabilidades: Deberá validar formato y tipo de los datos al momento de ingresarlos, validar en base al campo rucuni si ya existe el registro a insertar, además comprobará que todos los campos se inserten.

.

Tipo: OFACUEP

Caso de uso: Ingresar Oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos

Postcondiciones: Datos validados

Guardar oferta

Nombre: Guardar oferta

Responsabilidades: Deberá guardar los datos de la Oferta académica en el repositorio Oferta académica

Tipo: OFACUEP

Caso de uso: Ingresar Oferta Académica

Notas:

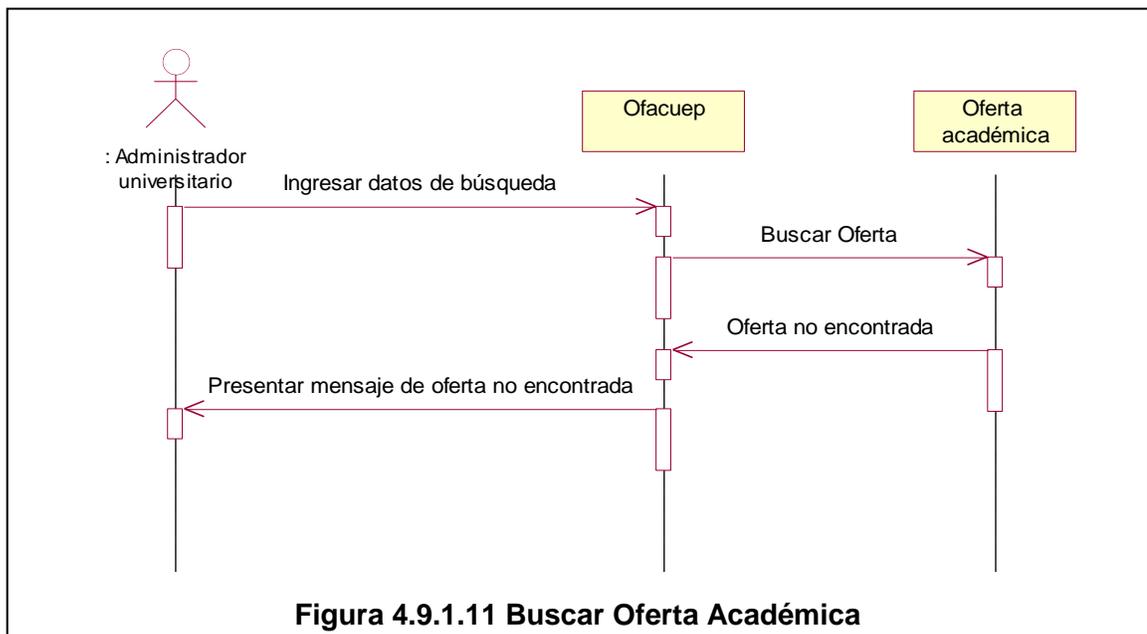
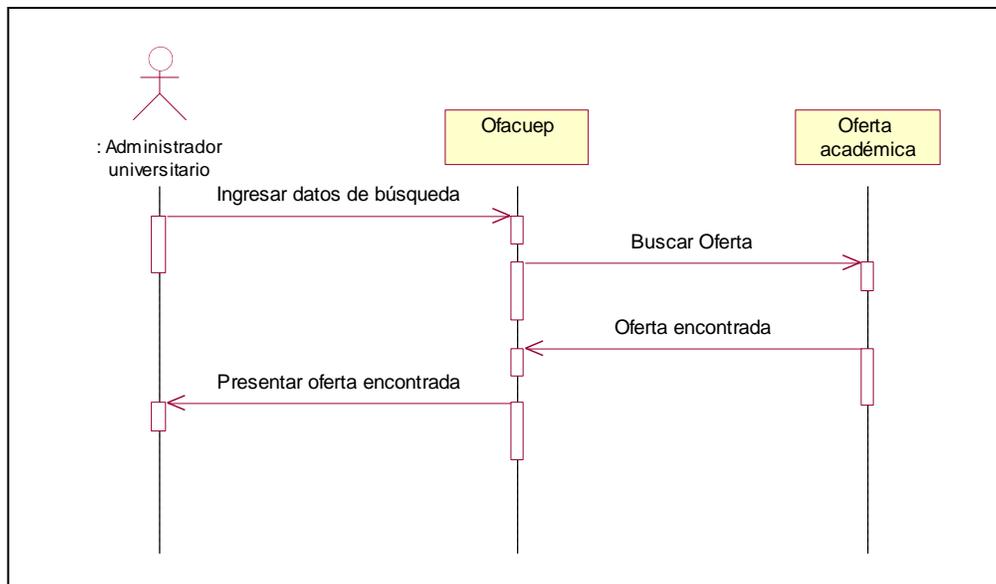
Excepciones:

Salidas:

Precondiciones: Validar datos

Postcondiciones: Oferta guardado

4.9.1.11 Buscar Oferta Académica



Buscar oferta

Nombre: Buscar oferta

Responsabilidades: Deberá buscar el registro de la Oferta académica de acuerdo a los datos de búsqueda

Tipo: OFACUEP

Caso de uso: Buscar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Ingresar datos de búsqueda

Postcondiciones: Oferta encontrada u Oferta no encontrada

Presentar oferta encontrada

Nombre: Presentar oferta encontrada

Responsabilidades: Deberá presentar en pantalla el registro de la oferta buscada

Tipo: OFACUEP

Caso de uso: Buscar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Buscar oferta

Postcondiciones:

Presentar msg oferta no encontrada

Nombre: Presentar msg oferta no encontrada

Responsabilidades: Deberá presentar en pantalla un mensaje indicando que no se encontró la oferta a buscar.

Tipo: OFACUEP

Caso de uso: Buscar oferta académica

Notas:

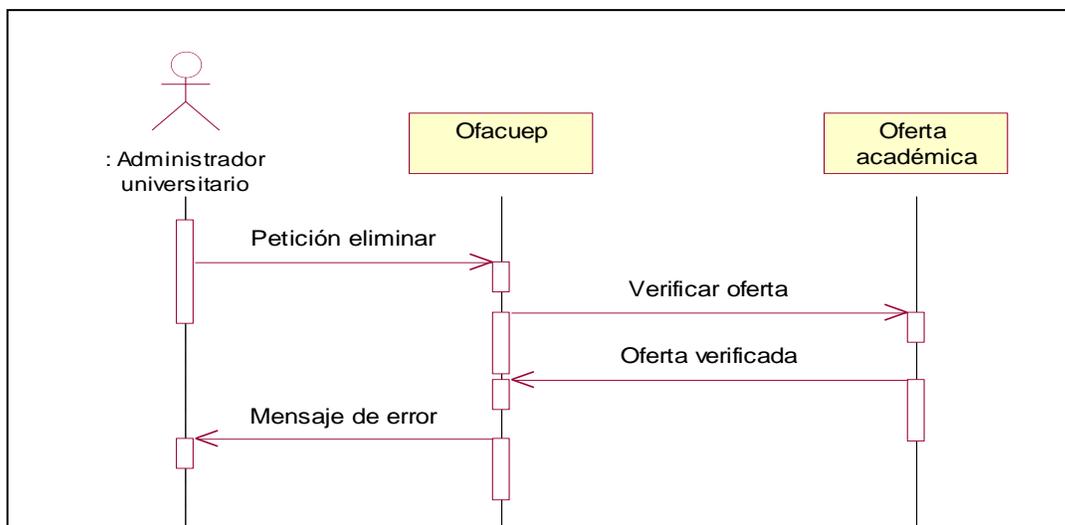
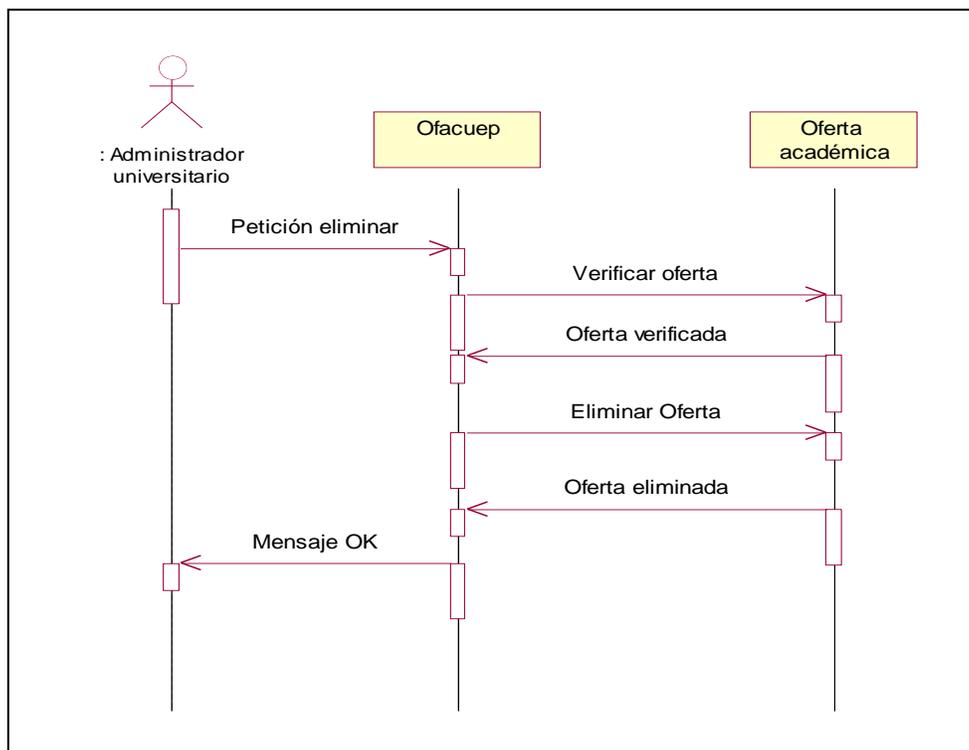
Excepciones:

Salidas:

Precondiciones: Buscar oferta

Postcondiciones:

4.9.1.12 Eliminar Oferta Académica



Verificar oferta

Nombre: Verificar oferta

Responsabilidades: Deberá verificar en base a codigoofer, si los datos de la oferta a eliminar tienen o no otros datos que dependan de ellos.

Tipo: OFACUEP

Caso de uso: Eliminar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Petición eliminar

Postcondiciones: Oferta verificada

Eliminar oferta

Nombre: Eliminar oferta

Responsabilidades: Deberá eliminar la oferta académica solicitada del repositorio Oferta académica.

Tipo: OFACUEP

Caso de uso: Eliminar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Verificar oferta

Postcondiciones: oferta eliminada

4.9.1.13 Modificar Oferta Académica

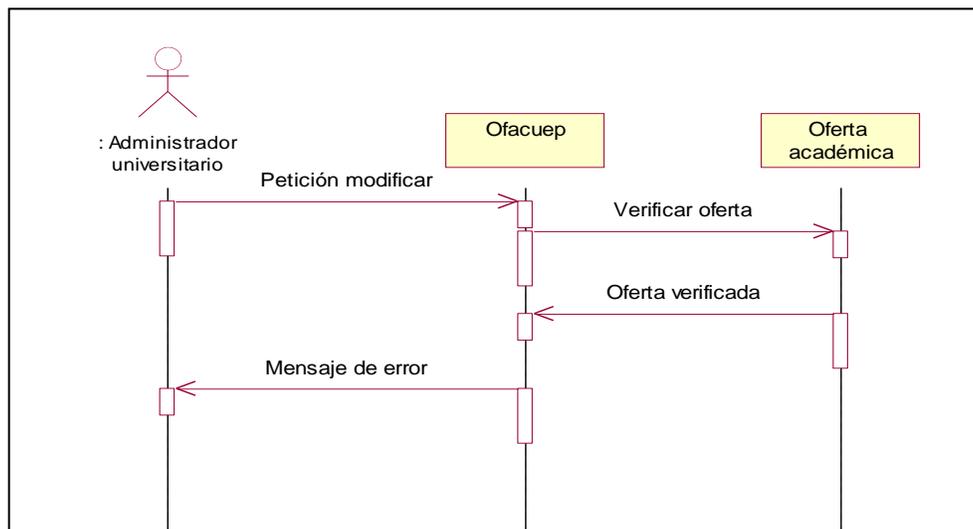
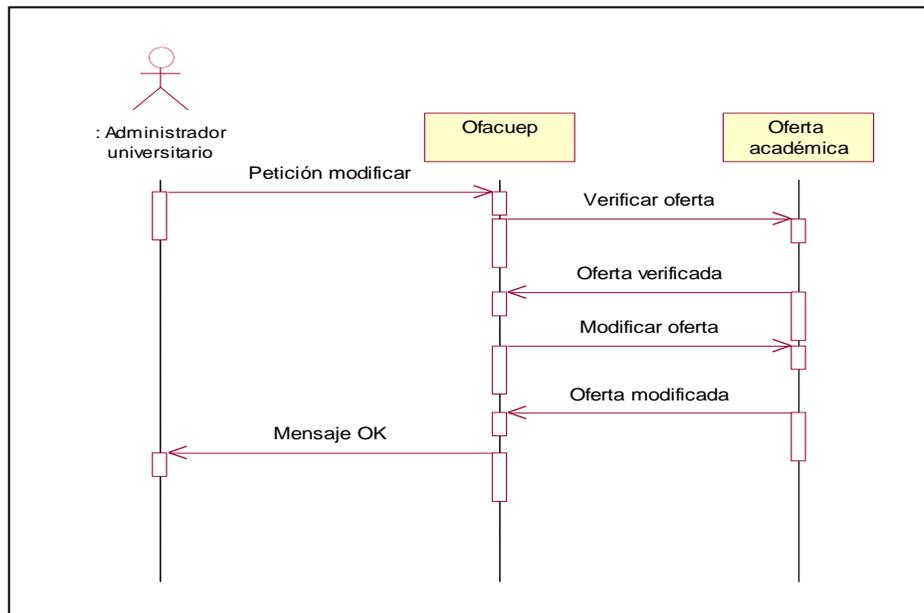


Figura 4.9.1.13 Modificar Oferta Académica

Verificar oferta

Nombre: Verificar oferta

Responsabilidades: Deberá verificar en base a codigoofer, si los datos de la oferta a eliminar tienen o no otros datos que dependan de ellos.

Tipo: OFACUEP

Caso de uso: Modificar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Petición modificar

Postcondiciones: Oferta verificada

Modificar oferta

Nombre: Modificar oferta

Responsabilidades: Deberá modificar la oferta académica solicitada del repositorio Oferta académica.

Tipo: OFACUEP

Caso de uso: Modificar oferta académica

Notas:

Excepciones:

Salidas:

Precondiciones: Verificar oferta

Postcondiciones: oferta modificada

4.9.1.14 Buscar Especialidades por el Usuario

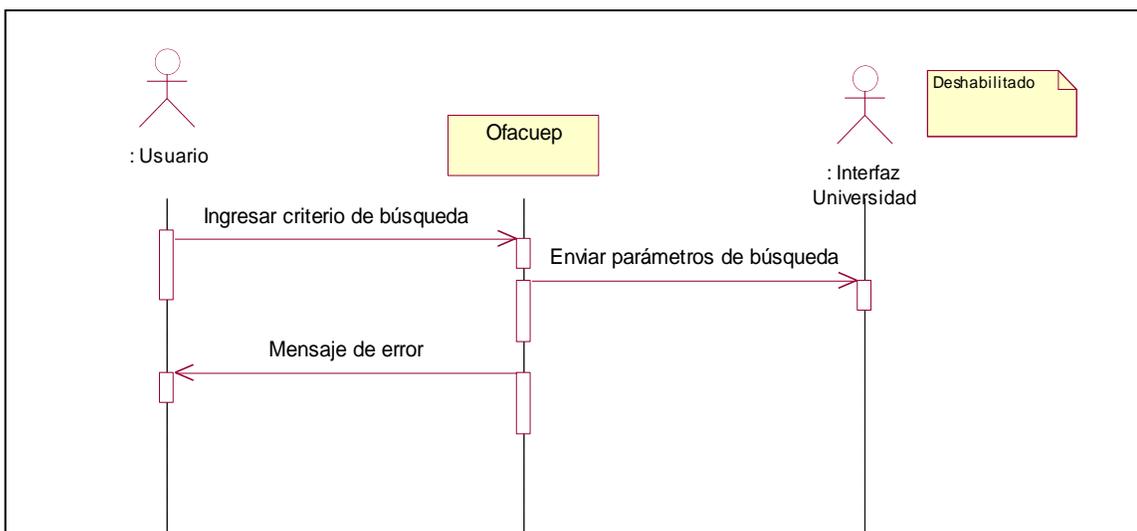
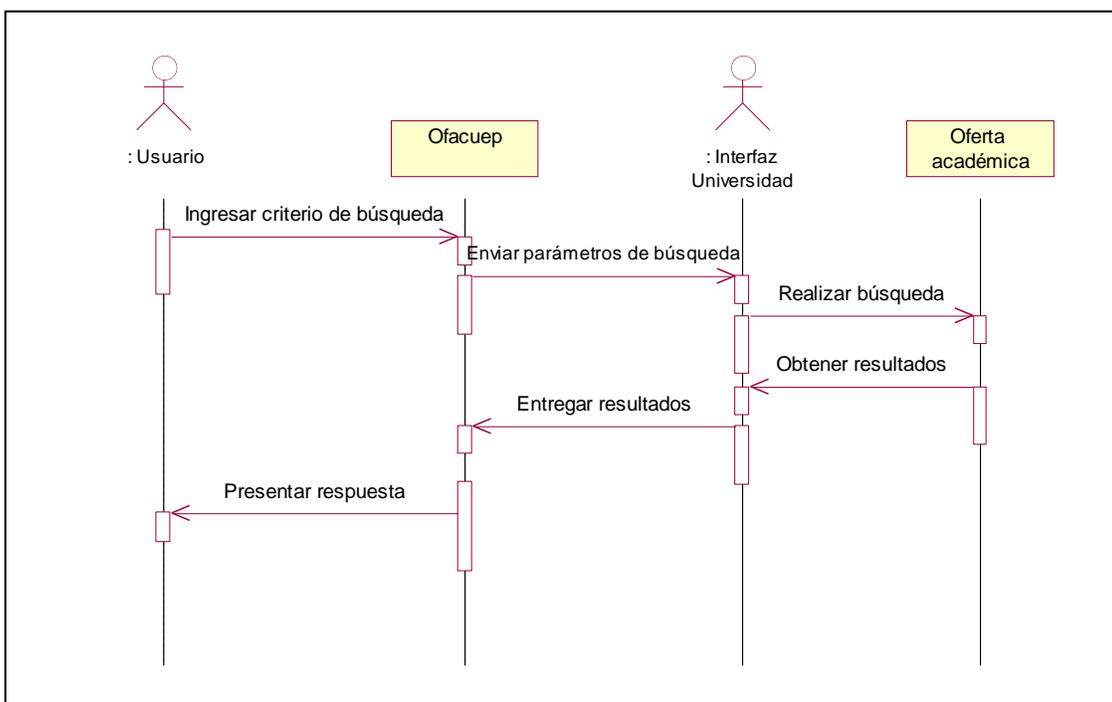


Figura 4.9.1.14 Buscar Especialidades por el Usuario

Realizar búsqueda

Nombre: Realizar búsqueda

Responsabilidades: La interfaz universidad (web service) deberá realizar la búsqueda en el repositorio Oferta académica de acuerdo a los parámetros de búsqueda

Tipo: OFACUEP

Caso de uso: Buscar especialidades por usuario

Notas:

Excepciones:

Salidas:

Precondiciones: Enviar parámetros de búsqueda

Postcondiciones: Obtener resultados

Presentar respuesta

Nombre: Presentar respuesta

Responsabilidades: Deberá presentar en pantalla el resultado de la búsqueda realizada

Tipo: OFACUEP

Caso de uso: Buscar especialidades por usuario

Notas:

Excepciones:

Salidas:

Precondiciones: Realizar búsqueda

Postcondiciones:

4.9.1.15 Buscar Especialidades por el Sistema Universidad

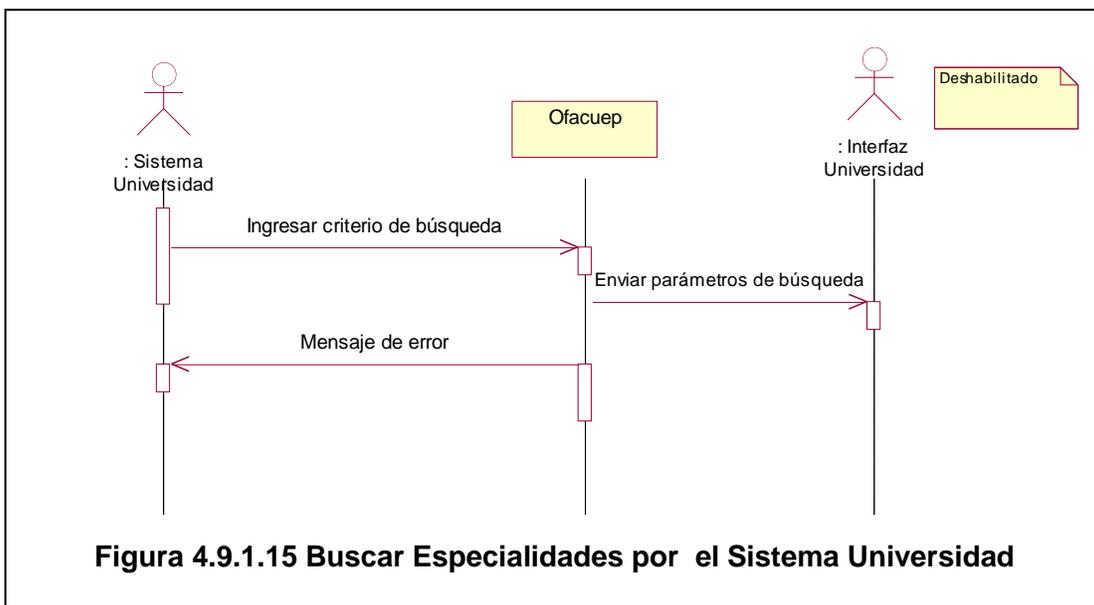
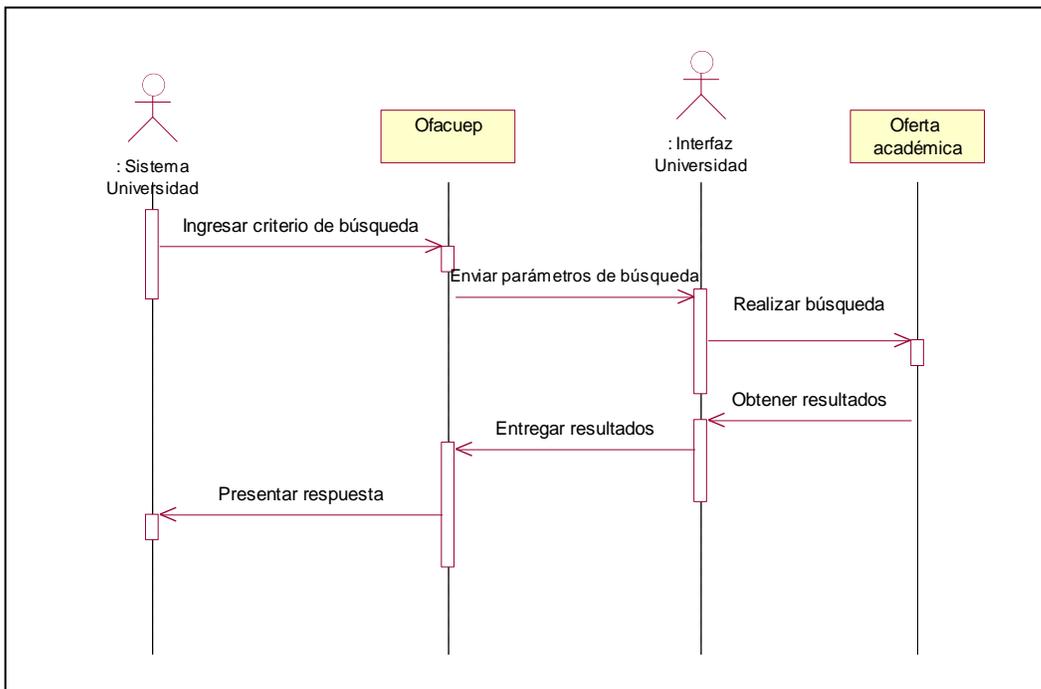


Figura 4.9.1.15 Buscar Especialidades por el Sistema Universidad

Realizar búsqueda

Nombre: Realizar búsqueda

Responsabilidades: La interfaz universidad (web service) deberá realizar la búsqueda en el repositorio Oferta académica de acuerdo a los parámetros de búsqueda

Tipo: OFACUEP

Caso de uso: Buscar especialidades por sistema universidad

Notas:

Excepciones:

Salidas:

Precondiciones: Enviar parámetros de búsqueda

Postcondiciones: Obtener resultados

Presentar respuesta

Nombre: Presentar respuesta

Responsabilidades: Deberá presentar en pantalla el resultado de la búsqueda realizada

Tipo: OFACUEP

Caso de uso: Buscar especialidades por sistema universidad

Notas:

Excepciones:

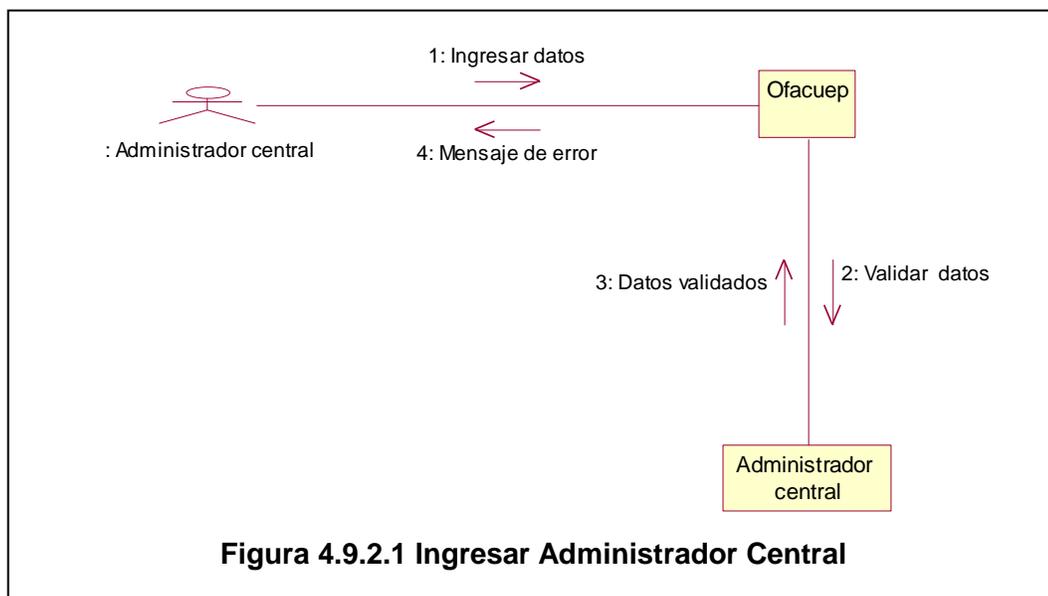
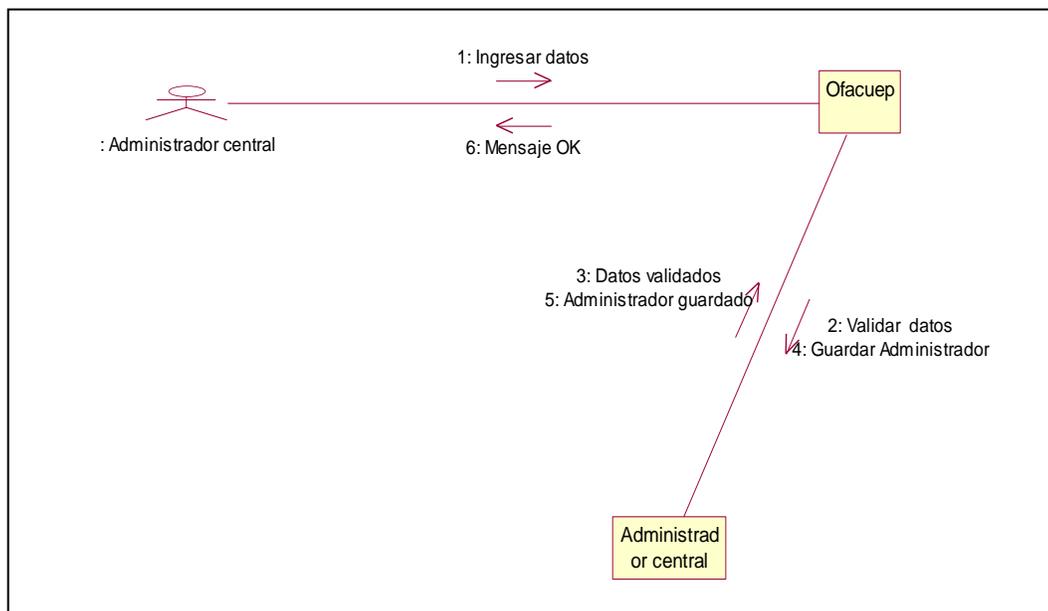
Salidas:

Precondiciones: Realizar búsqueda

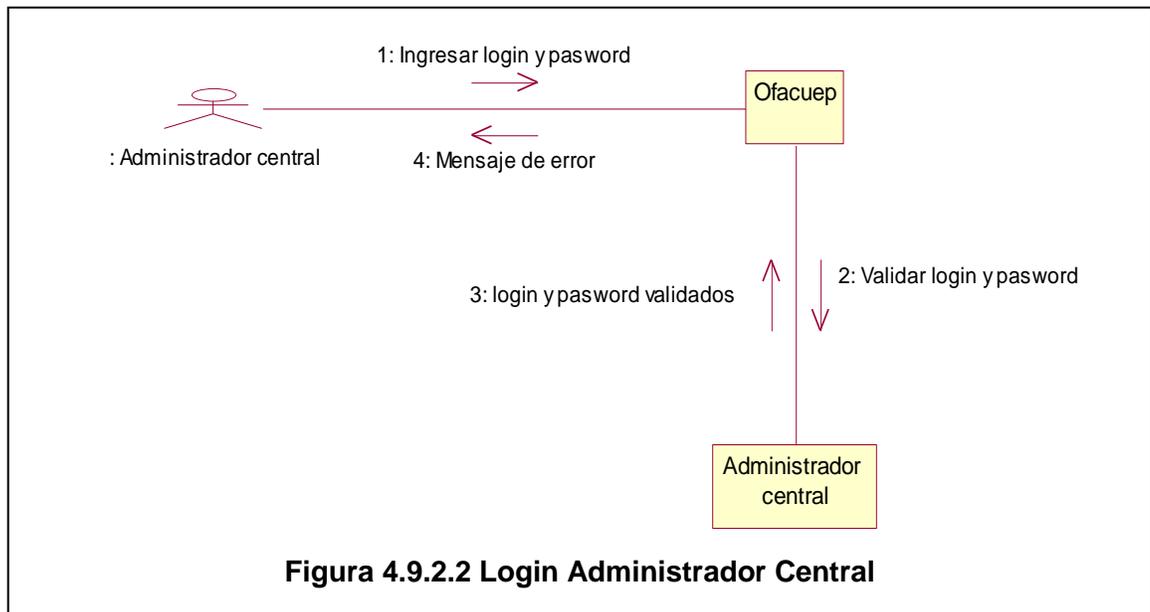
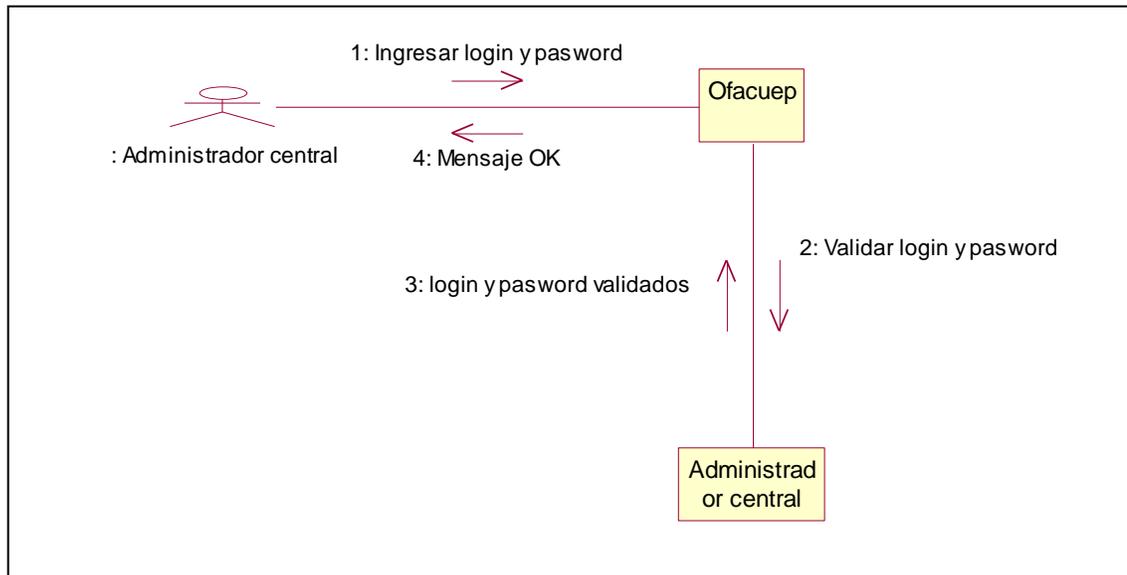
Postcondiciones:

4.9.2 DIAGRAMAS DE COLABORACIÓN

4.9.2.1 Ingresar Administrador Central



4.9.2.2 Login Administrador Central



4.9.2.3 Buscar Administrador Central

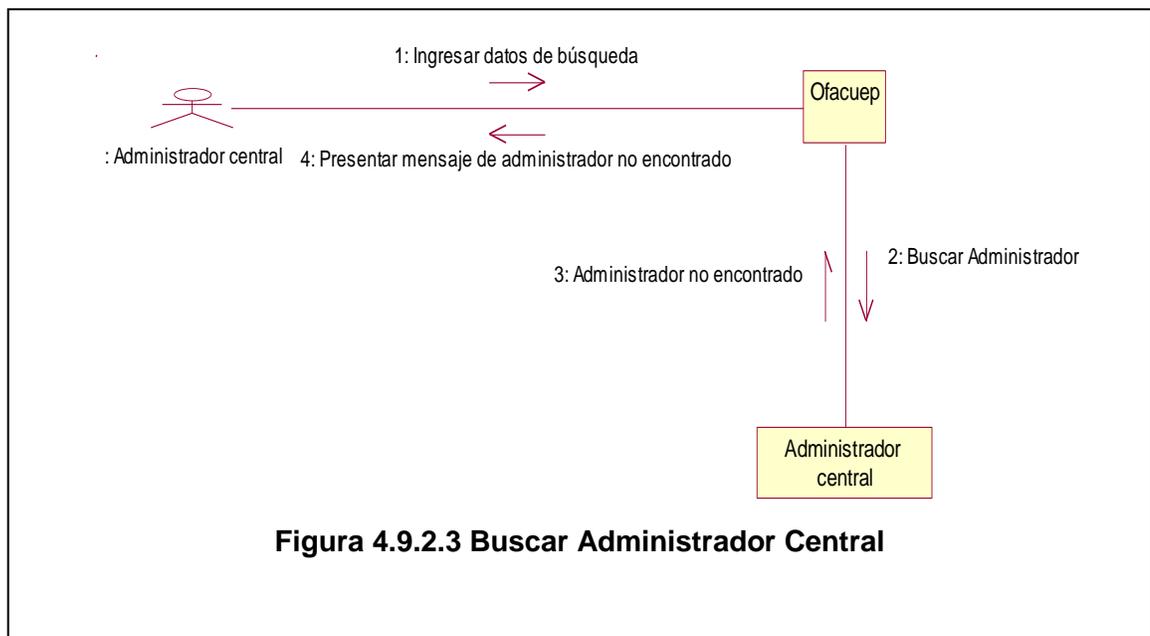
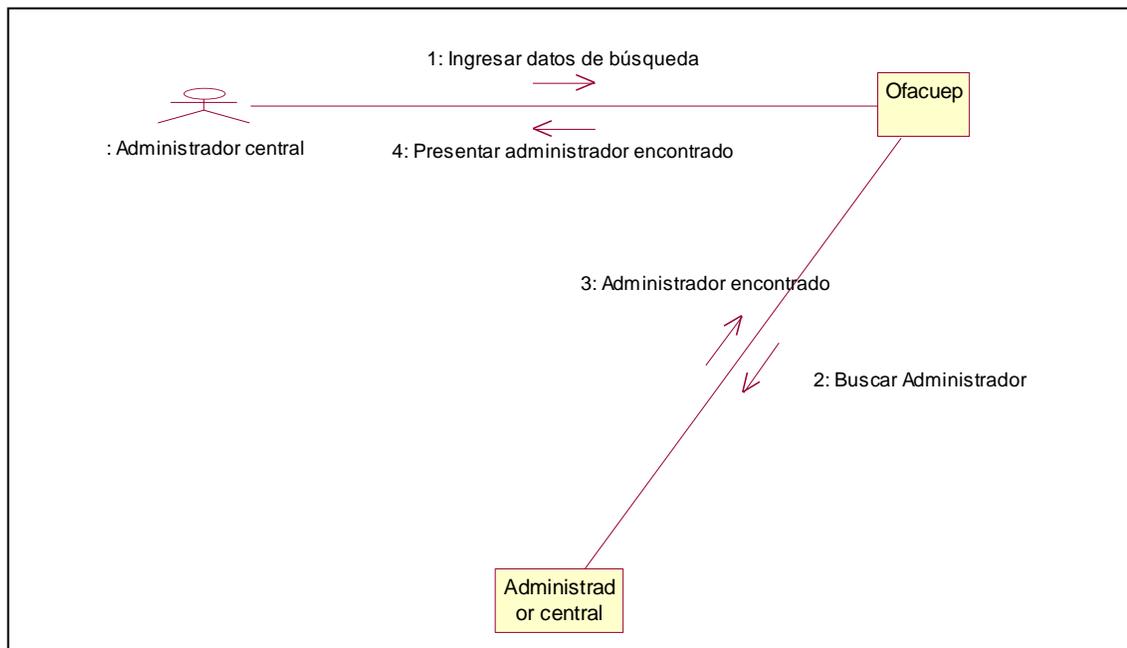
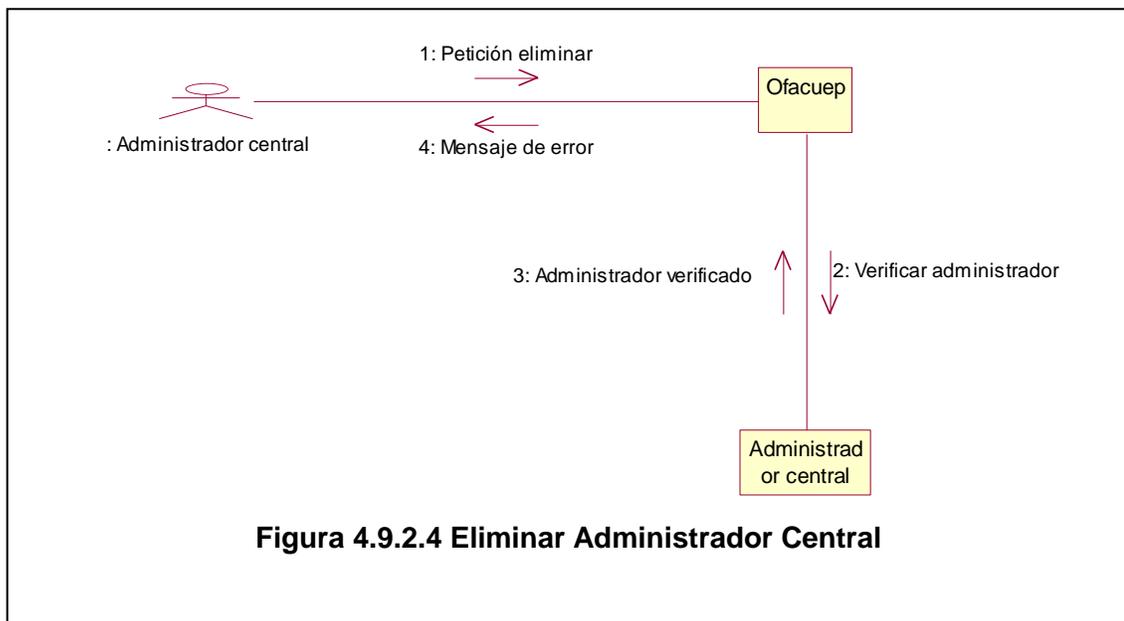
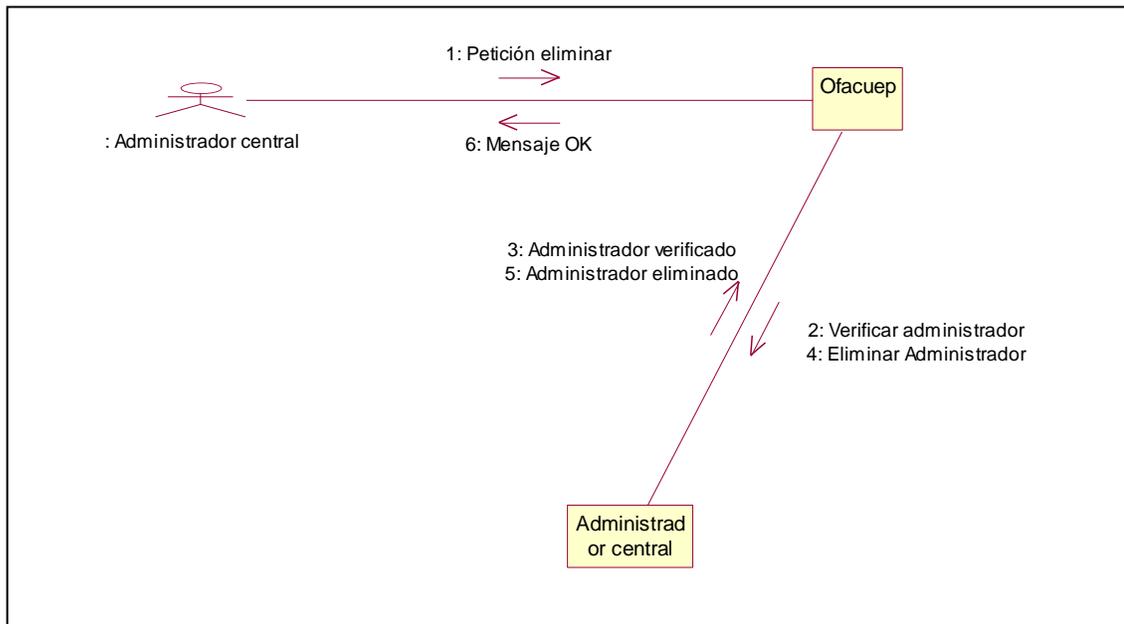
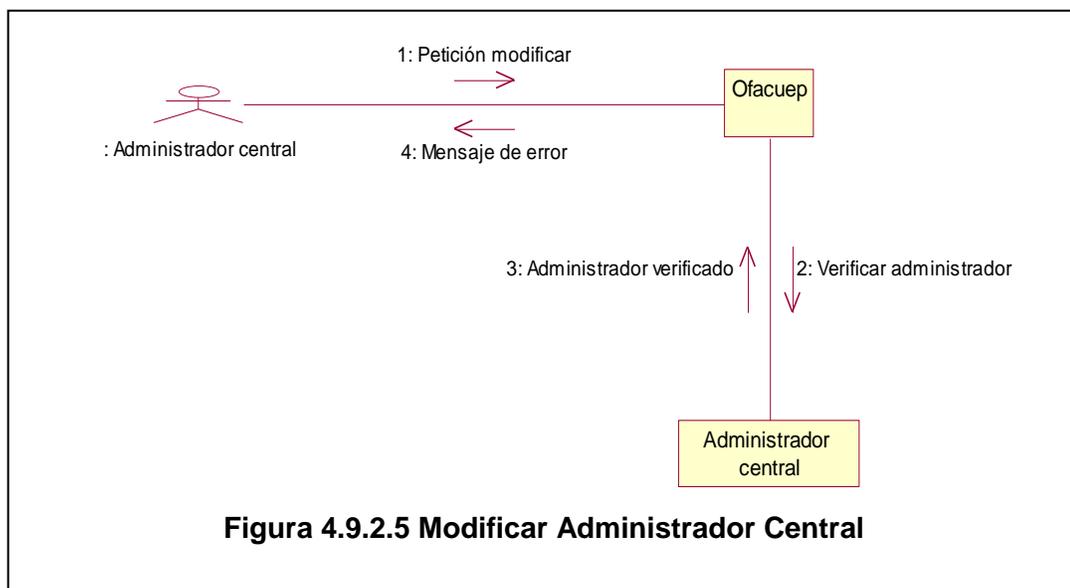
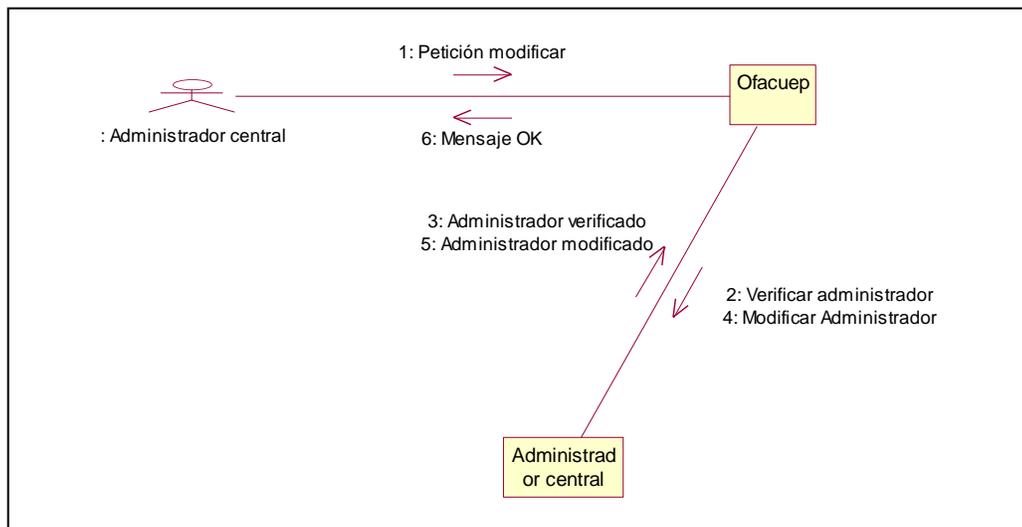


Figura 4.9.2.3 Buscar Administrador Central

4.9.2.4 Eliminar Administrador central



4.9.2.5 Modificar Administrador central



4.9.2.6 Suscribirse Universidad

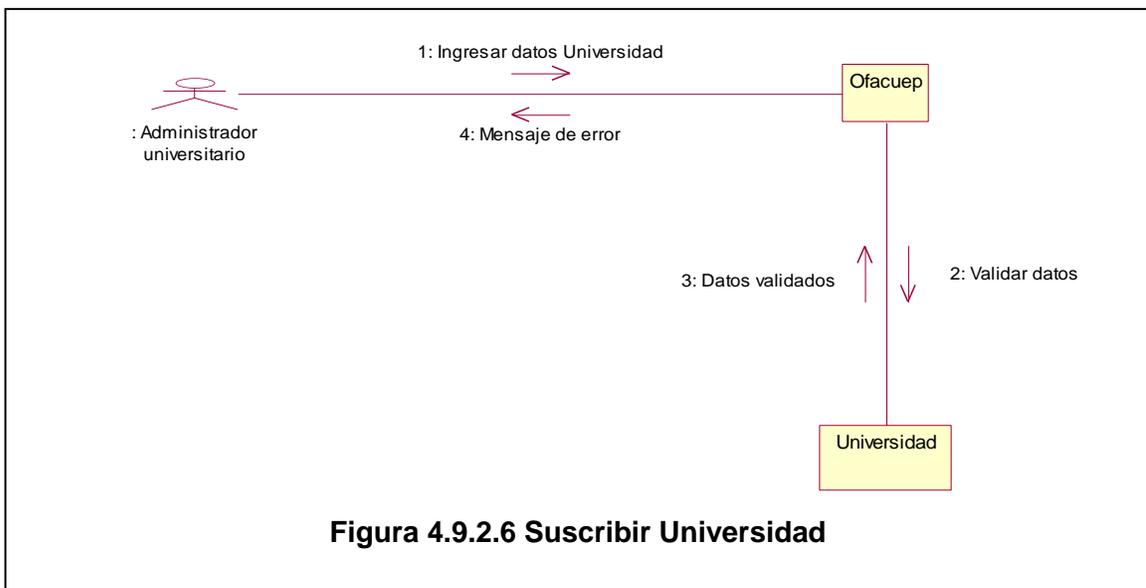
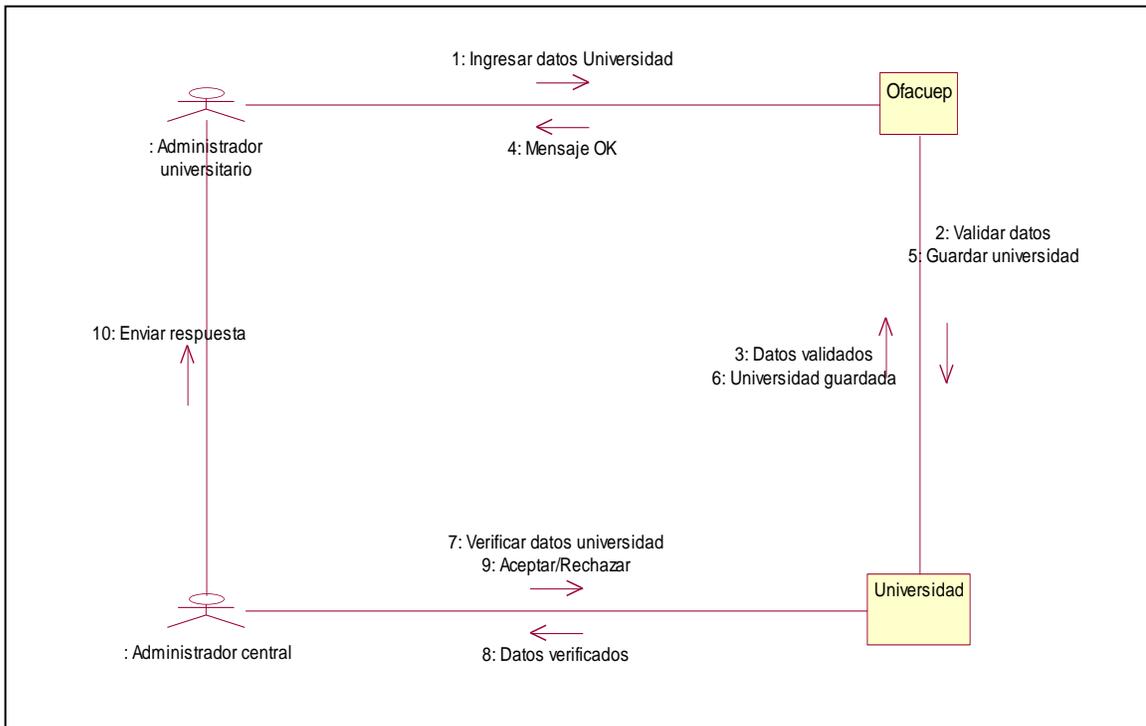


Figura 4.9.2.6 Suscribirse Universidad

4.9.2.7 Login Universidad

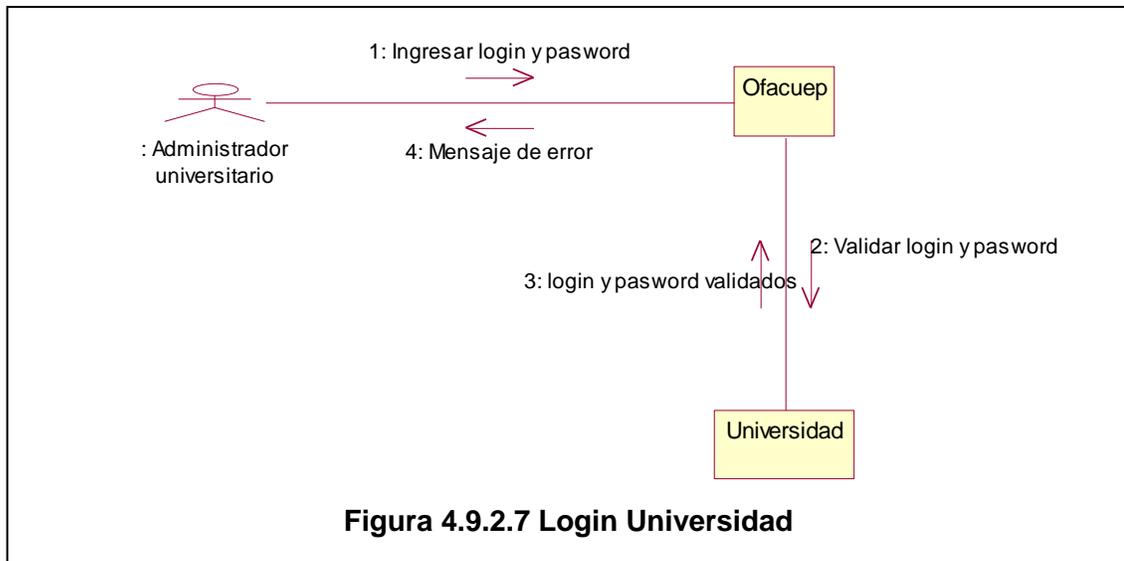
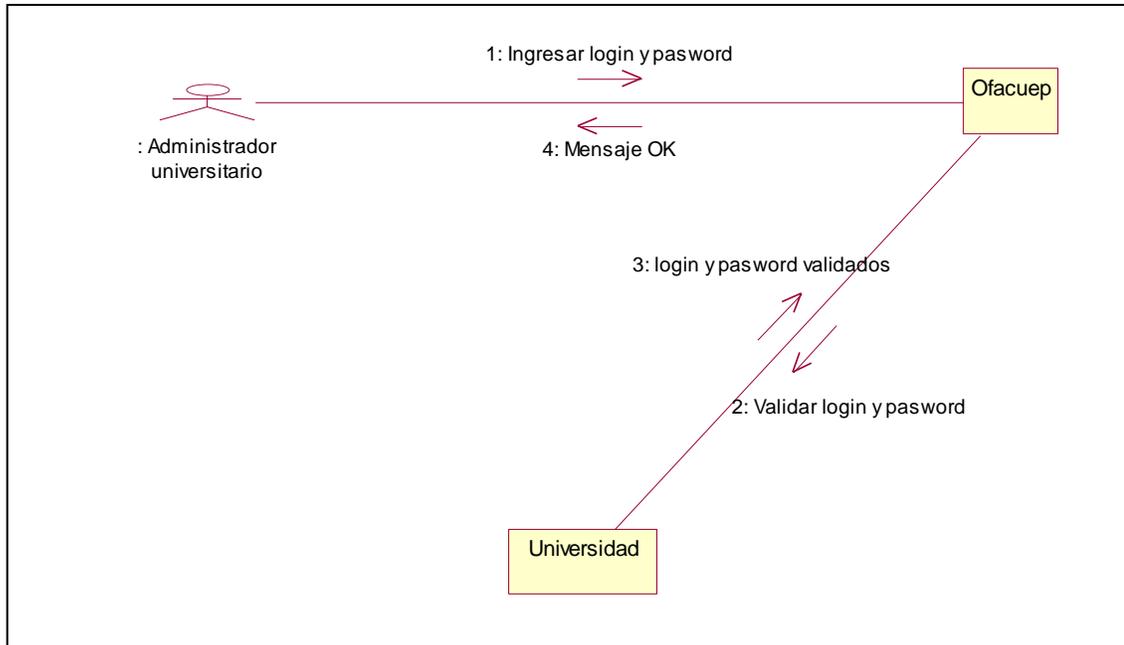


Figura 4.9.2.7 Login Universidad

4.9.2.8 Buscar Universidad

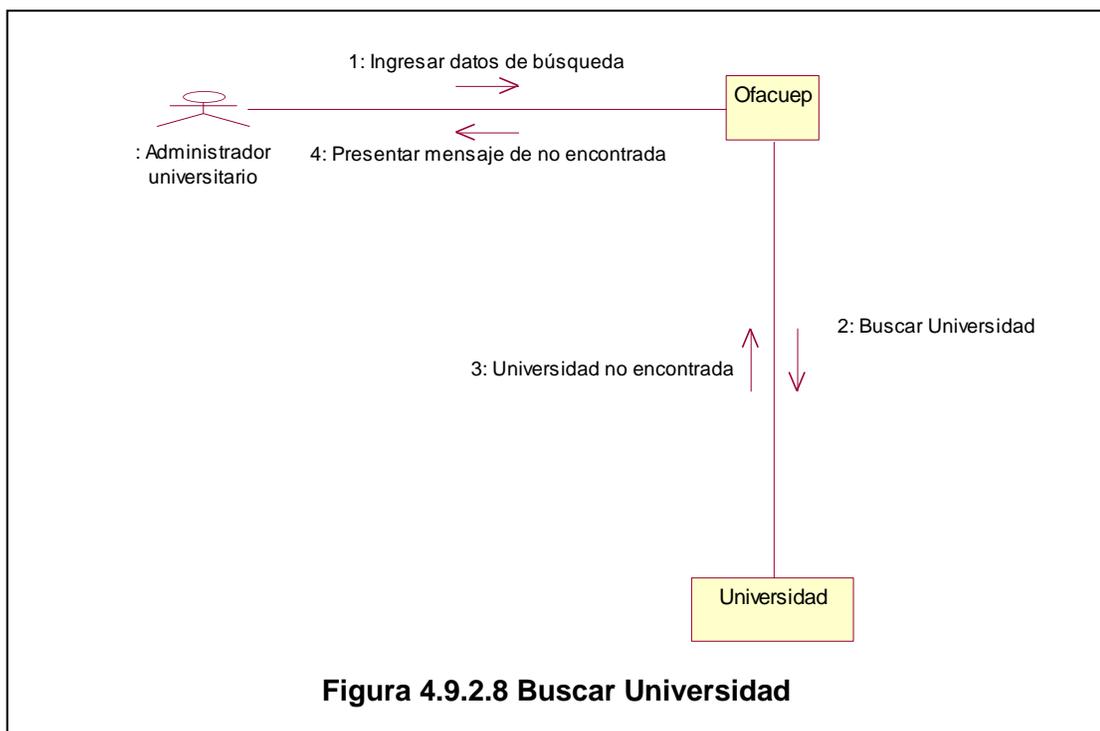
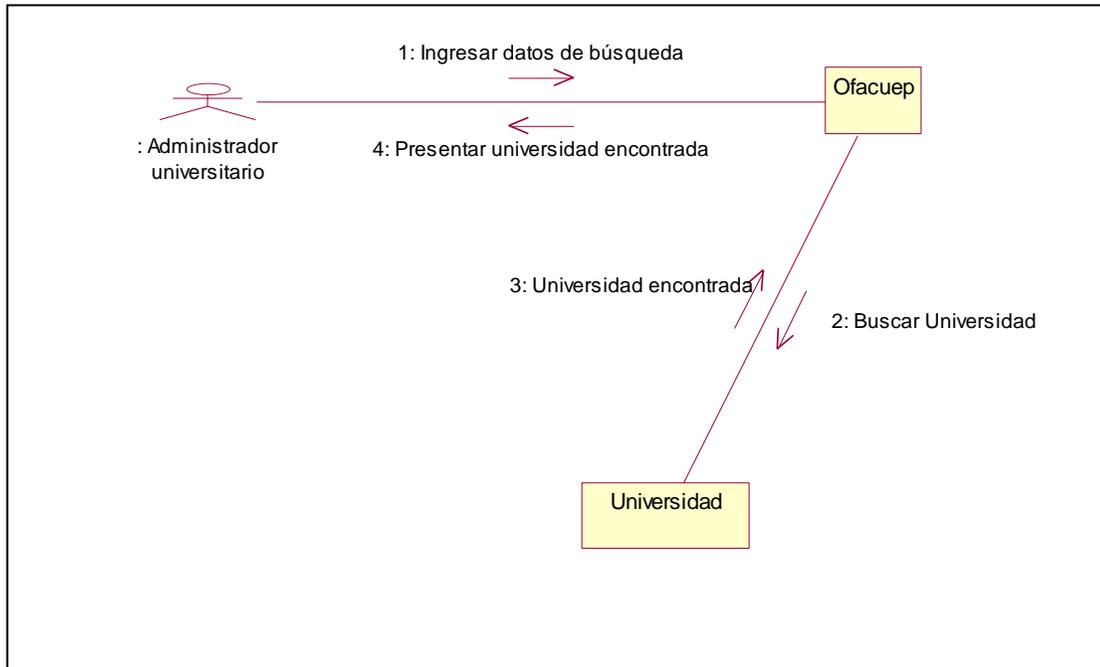


Figura 4.9.2.8 Buscar Universidad

4.9.2.9 Modificar Universidad

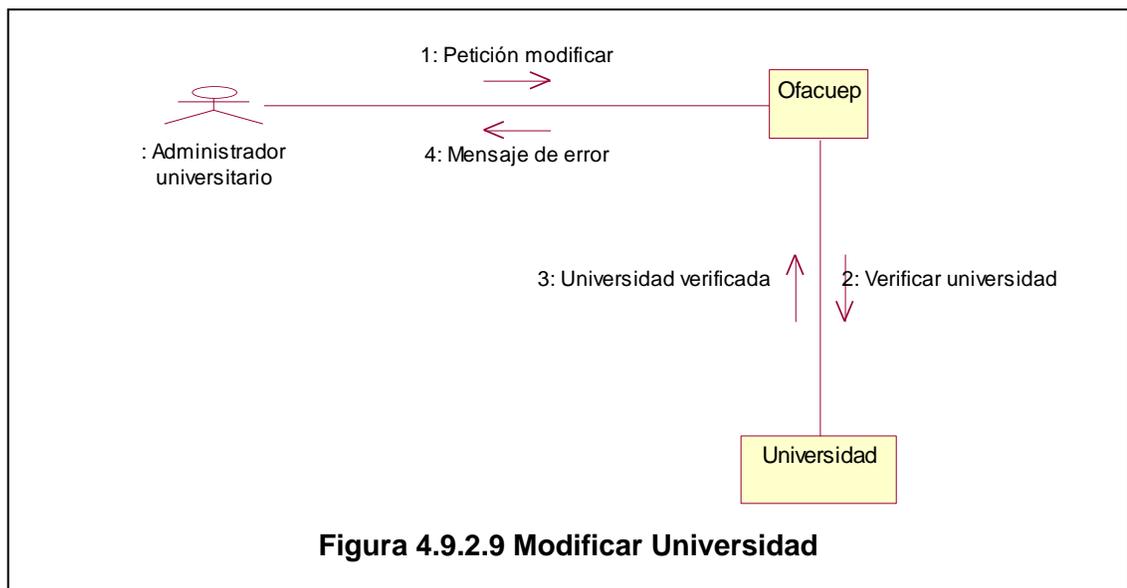
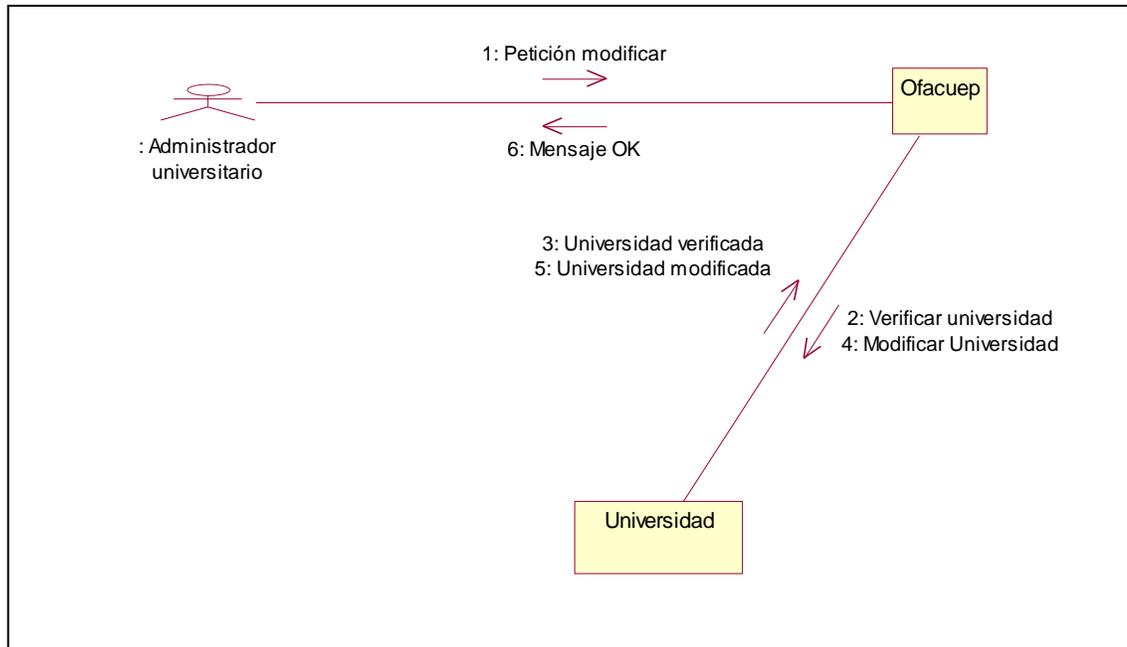
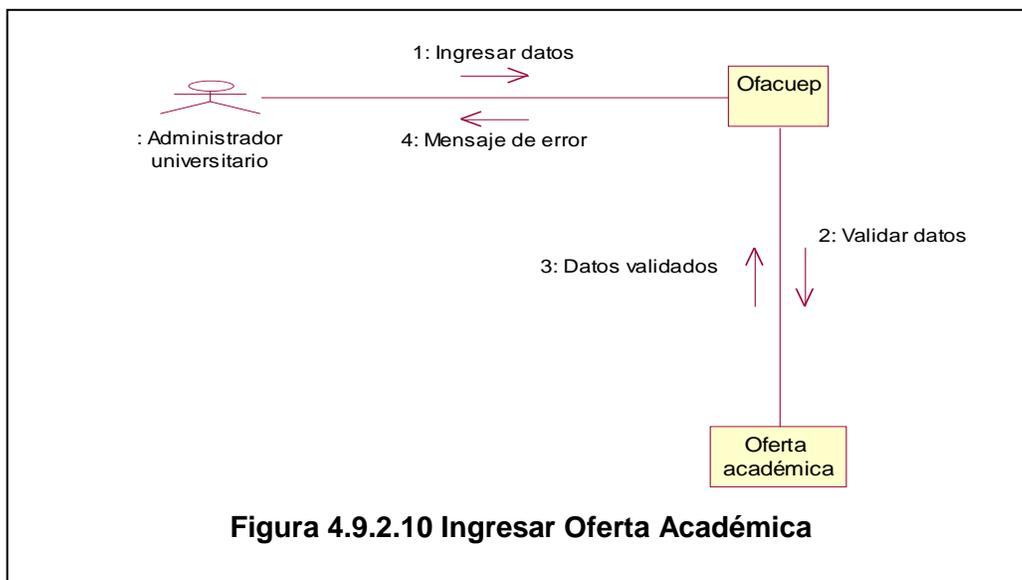
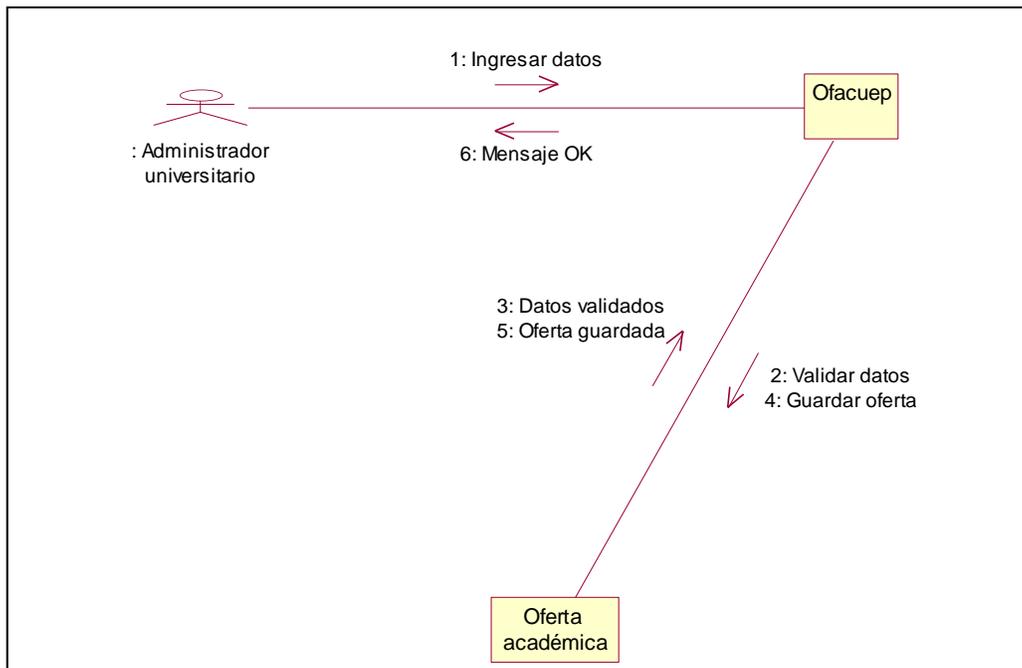
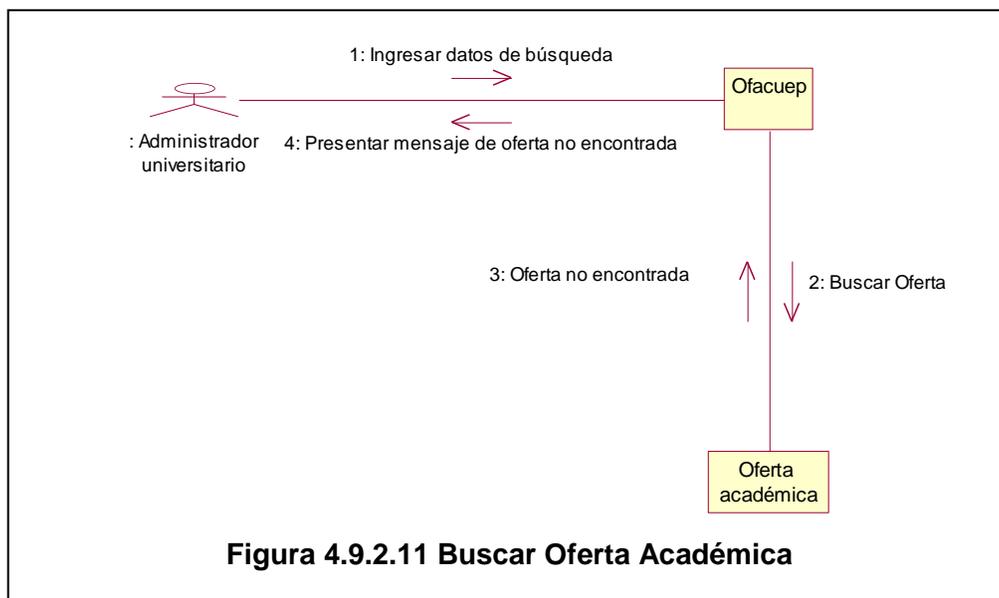
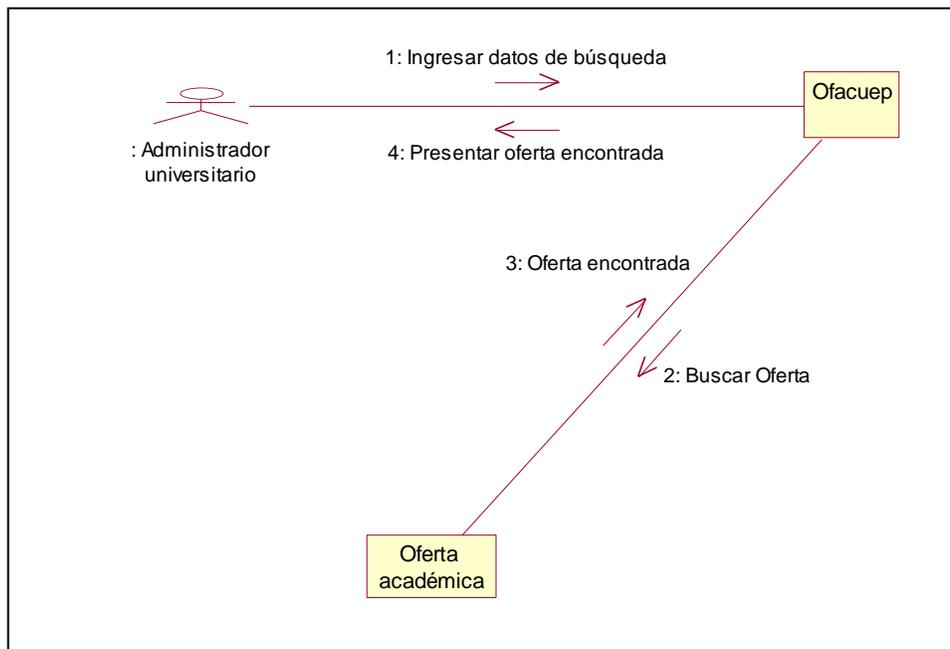


Figura 4.9.2.9 Modificar Universidad

4.9.2.10 Ingresar Oferta Académica



4.9.2.11 Buscar Oferta Académica



4.9.2.12 Eliminar Oferta Académica

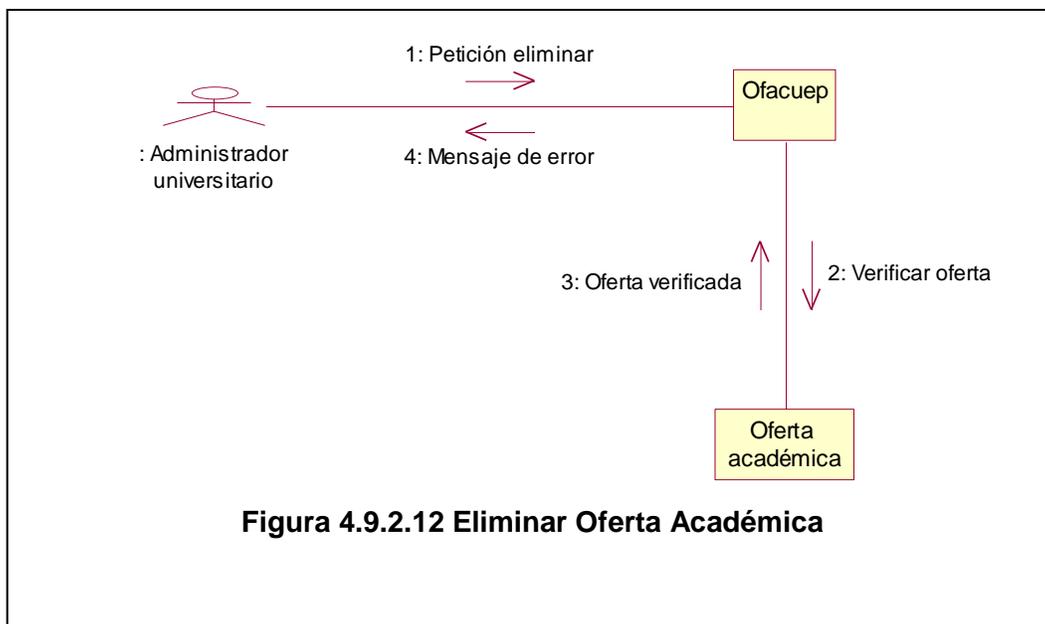
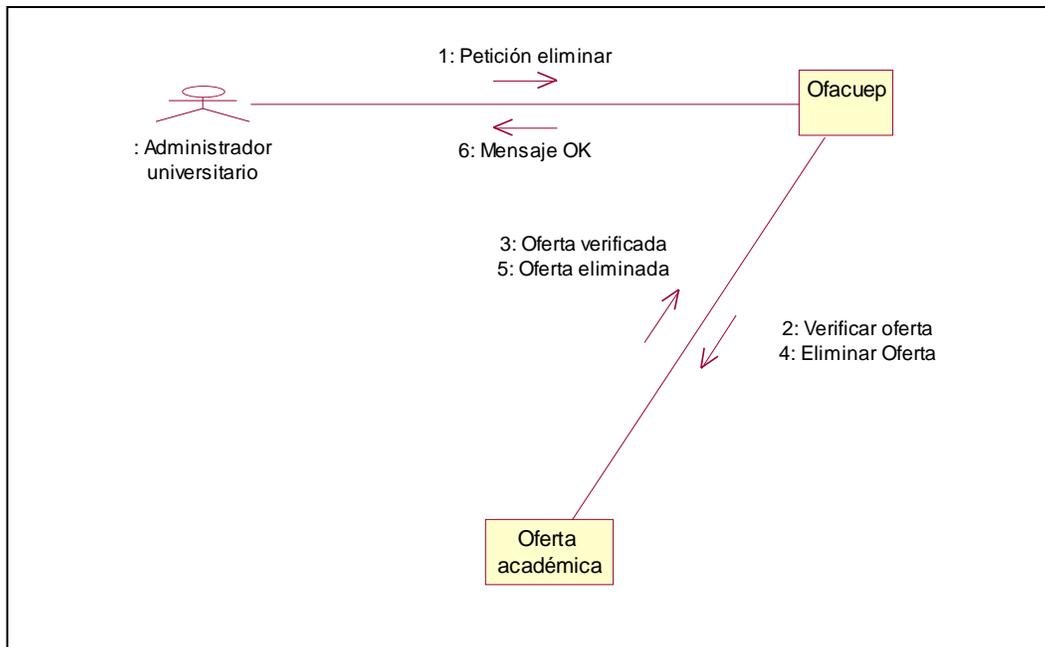


Figura 4.9.2.12 Eliminar Oferta Académica

4.9.2.13 Modificar Oferta Académica

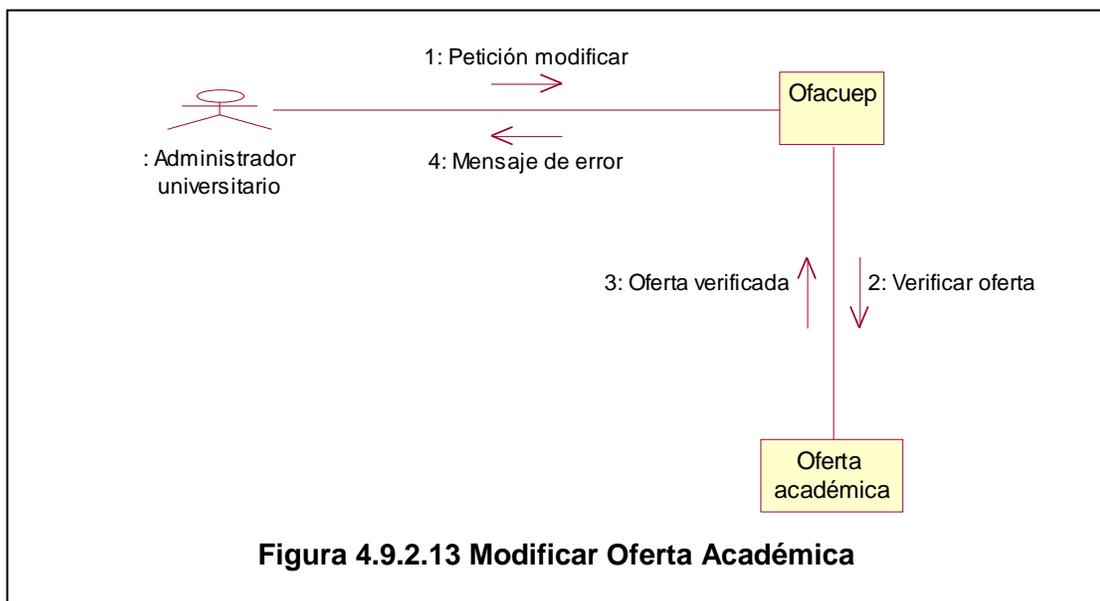
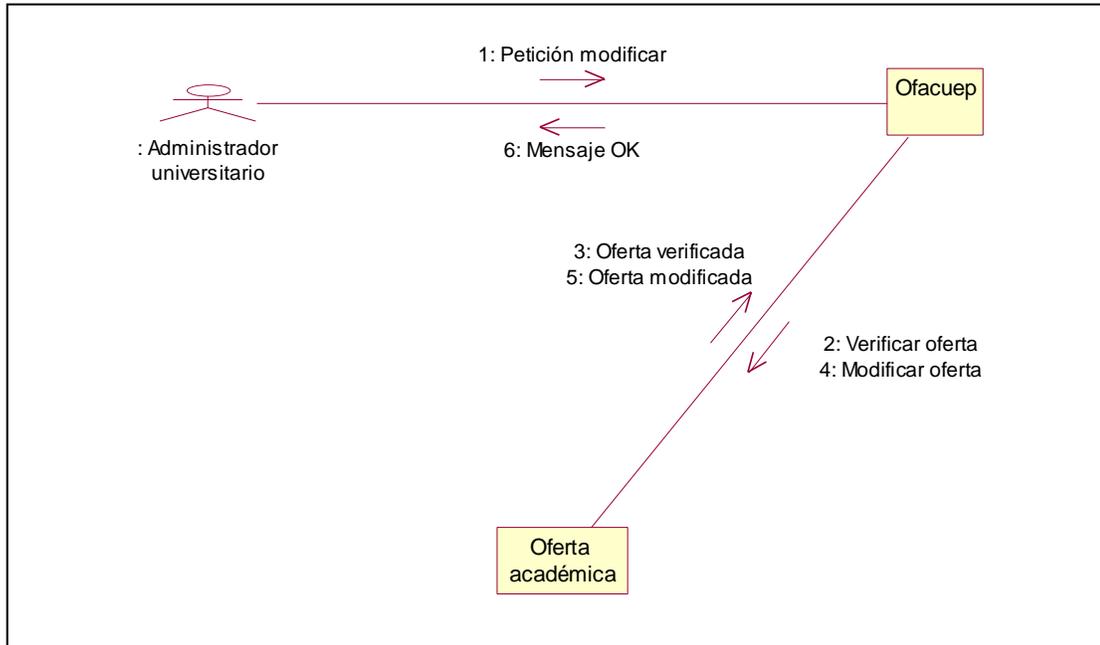


Figura 4.9.2.13 Modificar Oferta Académica

4.9.2.14 Buscar Especialidades por el Usuario

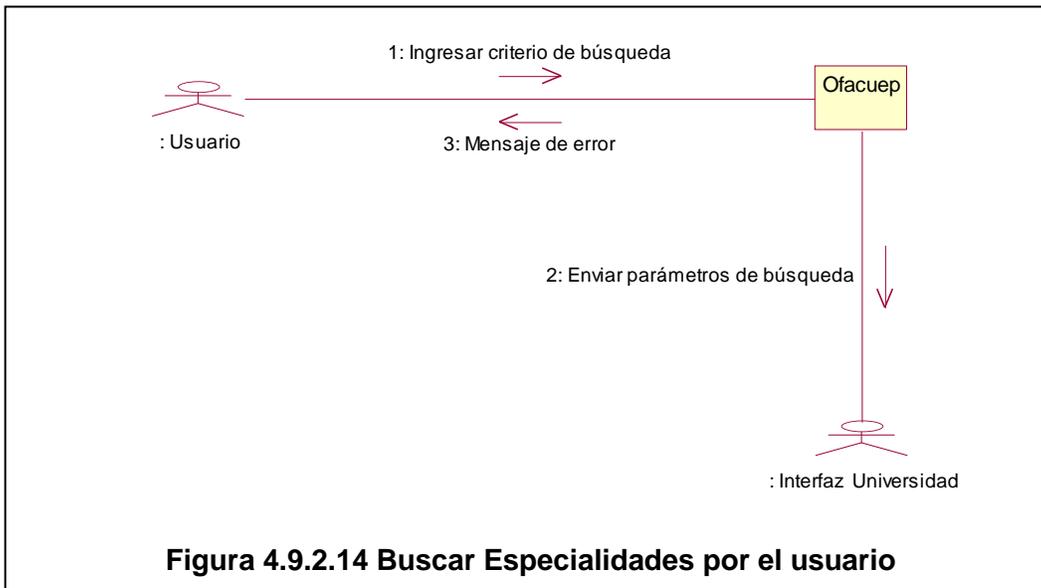
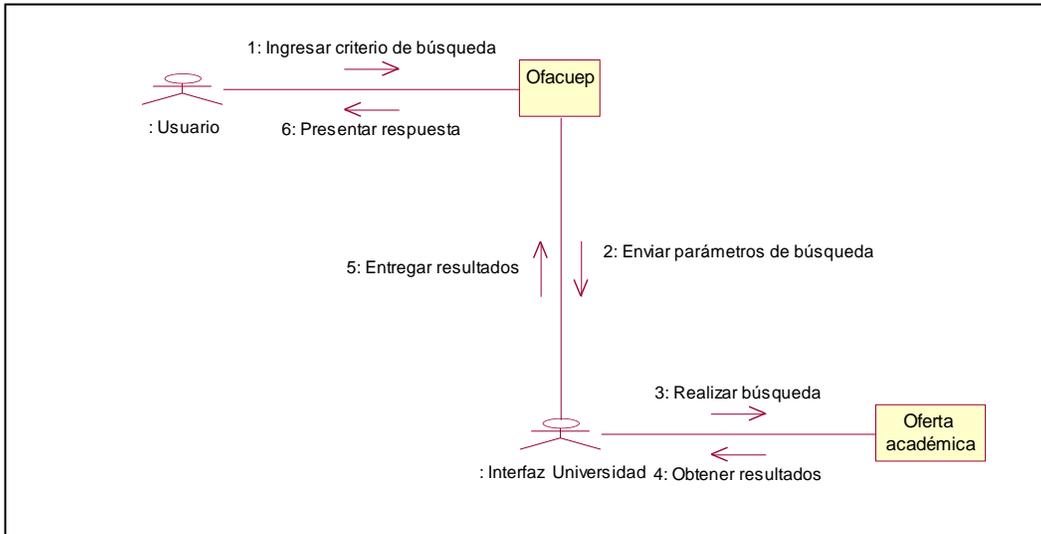
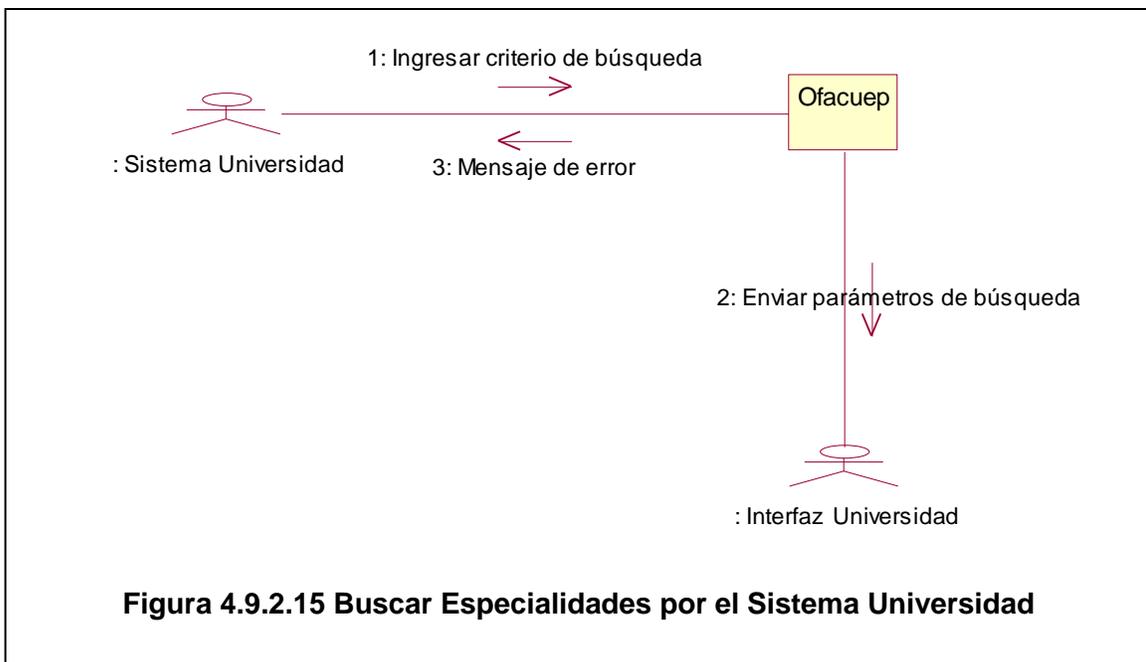
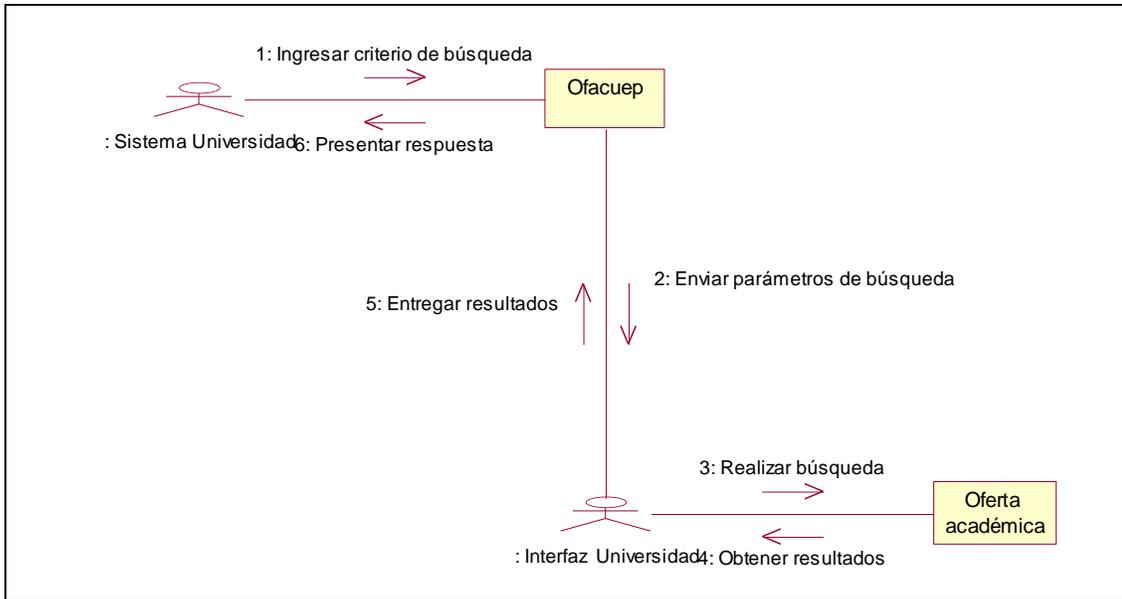
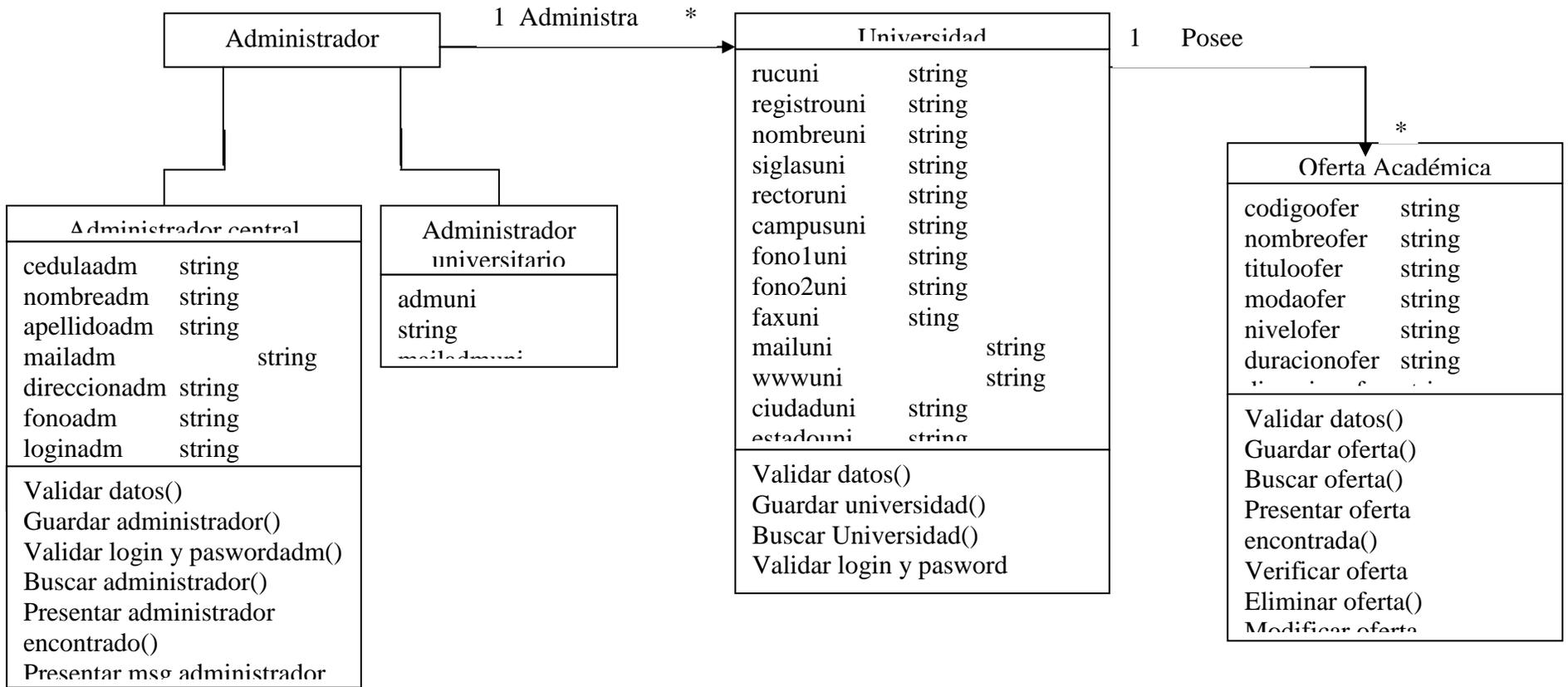


Figura 4.9.2.14 Buscar Especialidades por el usuario

4.9.2.15 Buscar Especialidades por el Sistema Universidad



4.9.3 DIAGRAMA DE CLASES



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Al finalizar el presente trabajo de investigación podemos concluir lo siguiente:

- Constatamos claramente cómo a través de los Web Services se comunican los ordenadores entre sí mediante XML, sin la intervención total del programador, y siendo completamente transparente para el usuario.
- Actualmente en el mundo del Internet, lo que está ganando espacio es la "Sencillez y la apertura", es decir disponibilidad para todos, por lo que la investigación que aquí ha sido expuesta, resulta de gran utilidad para la comunidad informática, debido a que conjuga sencillez con apertura y disponibilidad, XML con software libre.
- En el desarrollo de esta investigación no se utilizó base de datos alguna, bastó con guardar la información en archivos XML, por lo que podemos afirmar que las utilidades del XML son muy amplias, tal es así que en esta tesis el XML sustituyó a una Base de Datos.
- El uso de software libre utilizado en esta investigación como linux y php, fue muy importante a nivel económico y legal.

Económico porque los costos en cuanto a software y hardware fueron menores que si hubiésemos utilizado software propietario, ya que con una sola licencia de software, pudimos instalar en las máquinas necesarias, y sin requerimientos excesivos de hardware para su funcionamiento.

Legal ya que el software libre es siempre legal, salvo cuando se le imponen restricciones.

- Siendo la primera vez que programamos en php, podemos afirmar que su sintaxis es intuitiva y sencilla de aprender incluso para programadores sin mucha experiencia, pero a más de la sintaxis, en el desarrollo de esta investigación se tomaron en cuenta otros factores no tan sencillos y determinantes, como son las variables y configuraciones que maneja el servidor.
- Como todo trabajo investigativo implica cierto grado de esfuerzo y costo en capacitación, éste no fue la excepción, especialmente en XML con PHP y Web Services, sin embargo más que el gasto económico fue el esfuerzo empleado en capacitarnos, ya que por tratarse de una tecnología nueva, nos llevó más tiempo de lo previsto.
- La utilización de la metodología orientada a objetos de UML (Lenguaje de modelamiento unificado) en el Análisis y Diseño, fue de gran ayuda ya que desde el inicio pudimos visualizar claramente el sistema en sí.

5.2 RECOMENDACIONES

Al finalizar el presente trabajo de investigación recomendamos lo siguiente:

- Utilizar software libre como PHP y Linux, ya que a más de no necesitar licencia, son robustos, estables, y además no obligan a usar una plataforma en particular, como ocurre con otros tipos de software que no son libres.
- Tomar como base para futuras investigaciones este tema de tesis, pues su desarrollo es un aporte muy importante para la comunidad informática.

- Escoger desde el inicio una sola versión de PHP, ya que en algunos casos si se desea actualizar la versión, lo que se desarrolló en la anterior no sirve en la nueva.
- Utilizar una metodología orientada a objetos al desarrollar un software ya que representa un ahorro de tiempo, permitiendo tener un enfoque claro del software desde el principio.

CONTENIDO

<u>CAPÍTULO I:</u> APLICACIONES DISTRIBUIDAS.....	7
1.1. INTRODUCCIÓN.....	7
1.2. APLICACIONES DISTRIBUIDAS	8
1.3. COMUNICACIÓN EN LOS SISTEMAS DISTRIBUIDOS	10
1.4. CARACTERÍSTICAS PRINCIPALES DE UN SISTEMA DISTRIBUIDO.	11
1.5. MODELOS DE LAS APLICACIONES DISTRIBUIDAS	19
1.6. VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES DISTRIBUIDAS.	24
1.7. ASPECTOS A TENER EN CUENTA EN EL PROCESO DE INGENIERÍA.....	26
1.8. APLICACIONES DE LOS SISTEMAS DISTRIBUIDOS	31
<u>CAPÍTULO II:</u> SOFTWARE LIBRE U OPEN SOURCE.....	33
2.1. INTRODUCCIÓN.....	33
2.2. ORÍGENES.....	33
2.3. DEFINICIÓN DE SOFTWARE LIBRE.....	34
2.4. LINUX.....	40
2.5. PHP (PROFESIONAL HOME PAGES)	45

2.6. MYSQL	48
------------------	----

CAPITULO III: WEB SERVICES51

3.1. INTRODUCCIÓN.....	51
------------------------	----

3.2. DEFINICIONES DE WEB SERVICES.	53
---	----

3.3. WEB SERVICES.	54
-------------------------	----

3.4. SOAP (SIMPLE OBJECT ACCESS PROTOCOL, PROTOCOLO DE ACCESO SIMPLE A OBJETOS).....	61
--	----

3.5. WSDL (WEB SERVICES DESCRIPTION LANGUAGE, LENGUAJE DE DESCRIPCIÓN DE SERVICIOS WEB)	63
---	----

3.6. UDDI (UNIVERSAL DISCOVERY DESCRIPTION AND INTEGRATION , INTEGRACIÓN, DESCRIPCIÓN Y DESCUBRIMIENTO UNIVERSAL)	66
--	----

3.7. VENTAJAS Y DESVENTAJAS DE LOS WEB SERVICES.	68
---	----

CAPÍTULO IV: DESARROLLO DE LA APLICACIÓN OFERTA ACADÉMICA DE LAS UNIVERSIDADES DEL CENTRO DEL PAÍS -OFACUEP 70

4.1 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	70
---	----

4.2 FUNCIONES DEL SISTEMA.....	74
--------------------------------	----

4.3	REQUISITOS ESPECÍFICOS	77
4.4	REQUISITOS DE INTERFACES EXTERNAS	82
4.5	REQUISITOS DE DESARROLLO	82
4.6	REQUISITOS TECNOLÓGICOS DE HARDWARE Y SOFTWARE	82
4.7	ATRIBUTOS	83
4.8	FASE DE ANÁLISIS	84
4.9	FASE DE DISEÑO	114

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....
..151

<u>5.1</u>	<u>CONCLUSIONES</u>	<u>151</u>
<u>5.2</u>	<u>RECOMENDACIONES</u>	<u>152</u>

REFERENCIAS BIBLIOGRÁFICAS

- TURBAN, Efrain. McCLEAN, Ephraim. WETHERBE, James. "Tecnologías de información para la administración" Edit. CECSA.
- STOUT, Rick. "World Wide Web". Edit. McGraw Hill
- SHORT, Scott. "Creare XML Web Services", Edit. Mondori Informática.
- HAHN, Harley. "Internet Manual de Referencia 2° Edición". Edit. McGraw Hill
- BUYENS, Jim. "Aprenda Desarrollo de Base de Datos Web YA". Edit. McGraw Hill
- KAHN, Louis. LOGAN, Laura. "Construya su propia Web". Edit. McGraw Hill
- YOUNG, Michael J. "Aprenda XML ya". Edic. I. Edit: Microsft. 2000

DIRECCIONES WEB

- <http://dmi.uib.es/~bbuades/sistdistr/sistdistr.ppt>
Gabriel Buades .Ingeniería del Software III. Enero 1999
- <http://www.revista.unam.mx/vol.2/num1/art3/>
Revista Digital Universitaria. Vol.2 No.1 Rolando Menchaca Méndez.31 de Marzo de 2001.
- <http://www.inst-informatica.pt/v20/ibst/estandar/8/ibdce.htm>
Open Software Foundation. The OSF Distributed Computing Environment: End User Profiles.2004
- <http://www.elai.upm.es/spain/Investiga/GCII/areas/administracion/DCOM.htm#Introducción>
Microsoft COM Component Object Model.2004
- <http://manifiesto.cofradia.org/introduccion.html>
Grupo de Usuarios de Linux de México. 2003
- www.gnu.org/philosophy/categories.es.html
Free Software Foundation.1999
- <http://www.desarrolloweb.com/articulos/304.php?manual=12>
Desarrollo WEB. 2003

- http://www.fcad.uner.edu.ar/Extension/2_jor_adm_inf/Ponencias/Molinari/Arquitecturas.ppt

Universidad Nacional de la Plata.Laboratorio de la nuevas tecnologías Informáticas.2003

- <http://www.desarrolloweb.com/articulos/450.php?manual=18>
Desarrollo WEB. 2003

- http://www.fisica.uson.mx/carlos/WebServices/WS_UDDI.htm

MC Carlos Lizárraga Celaya. Departamento de Física. Universidad de Sonora. Abril 2002

- <http://www.arachnida.com/attach/78/default/Serviciosdetecnolog%C3%A4Daweb.doc>

Ramírez, Moisés. Servicios de tecnología web. Presidente de Arachnida / iKels. 2003

- <http://www.programacion.com/tutorial/xmlrpcsoap/>

Programación de castellano.2002

- <http://www.itrainonline.org/itrainonline/spanish/opensource.shtml>

Ittrain on line Compartiendo el conocimiento sobre internet.2003

- http://www.interplanet.com.mx/deinteres_linux_opensource.htm

Interplanet.2003

- <http://www.lacompu.com/notas/elphp/index.php3>

La compu.com Política de protección de datos.2003

- <http://web-services.bankhacker.com/>

Banhacker Wev services.2004

- <http://www.opensource.org/>

Open Source by the Open Source Initiative.2005

- <http://www.glosarium.com/term/781,14,xhtml>

David Carrero Fernandez-Baillo & Ferca. Diccionario Informático. 2004

- <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/indice.HTM>
Instituto Nacional de Estadística e Informática.2004
- http://productos.bvsalud.org/tutorials/es/preparadobvs_es/preparadobvs_es-3-4.html
TID/BIREME.Tutoriales preparados. 2004
- http://es.wikipedia.org/wiki/Sistema_operativo
David Carrero Fernandez-Baillo & Ferca. Wikipedia la Enciclopedia Libre. 2004.
- <http://www.microsoft.com/spain/partner/licencias/default.asp>
Microsoft Corporation.2005
- <http://www.java.sun.com>
Products & Technologies. Java Technology. 2005
- <http://www.soaplite.com>

ANEXO 1

Cd que contiene lo siguiente :

- **Archivo de la tesis** : “Aplicabilidad de la tecnología Web Services basada en software Open Source o de Código Abierto, orientada a la Oferta Académica de las Universidades del Centro del País”.
- Manual del Sistema OFACUEP

Anexo 2.1

Descripción /Operación a realizar	Linux	Windows
Tipo de software	Libre	Propietario
Archivos que pueden leer	Todos los archivos comerciables y no comerciables	Archivos comerciables
Seguridad	Dificultad para violar las seguridades	Fácil de violar para los hackers
Rapidez	Mayor estabilidad y más eficiencia en su código fuente, por tanto más rápido	Menos rapidez
Costos de mantenimiento	Menor costo	Mayor costo
Costos en software	Gratuito	Costos altos
Propenso a contraer virus		Mayor seguimiento de virus

Restricción de tráfico	IP relay	
Reproductores midi karaoke		Van basco
Editores simples de gráficos	Kpaint, <u>Tuxpaint</u> , Xpaint, <u>Gpaint</u> , Killustrator	Paint
Convertir los archivos Gráficos a pseudográficos	Aalib.	
Bases de Datos Locales	<u>KNoda</u> , <u>Gnome DB Manager</u> , <u>OpenOffice</u> , <u>MySQL</u> , InterBase, BDB, <u>Rekall</u> , <u>StarOffice</u> , Adabase.	Access
Email + groupware	CommuniGate Pro, Bynari's Insight GroupWare Suite, Samsung Contact, Teamware Office para Linux, Novell Netmail, Amphora. Basado en zope y qmail, Tutos. Based on Linux / Apache / PHP / Mysql / Sendmail, Kroupware, SuSe Linux Openexchange Server, phpGroupWare, SCOffice Mail Server	Microsoft exchange
Server / router en un diskette.	Tantos como se desee con : <u>muLinux</u> , <u>Dachstein</u> , <u>Serverdisk</u> , <u>Fli4l</u> .	Imposible